Membrane Systems Combining Variable Molecular Structures with Discretised Reaction Kinetics

From a Toy to a Tool in Systems Biology

Thomas Hinze

Friedrich-Schiller University Jena
Department of Bioinformatics at School of Biology and Pharmacy

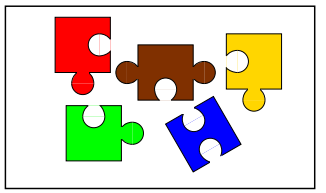
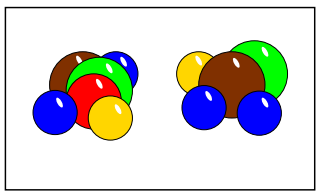
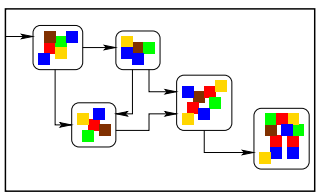
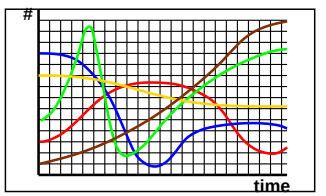
thomas.hinze@uni-jena.de

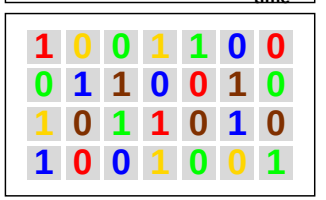
November 25, 2009

Outline

CSMs with Incomplete Protein Activation Information: KaiABC Oscillator

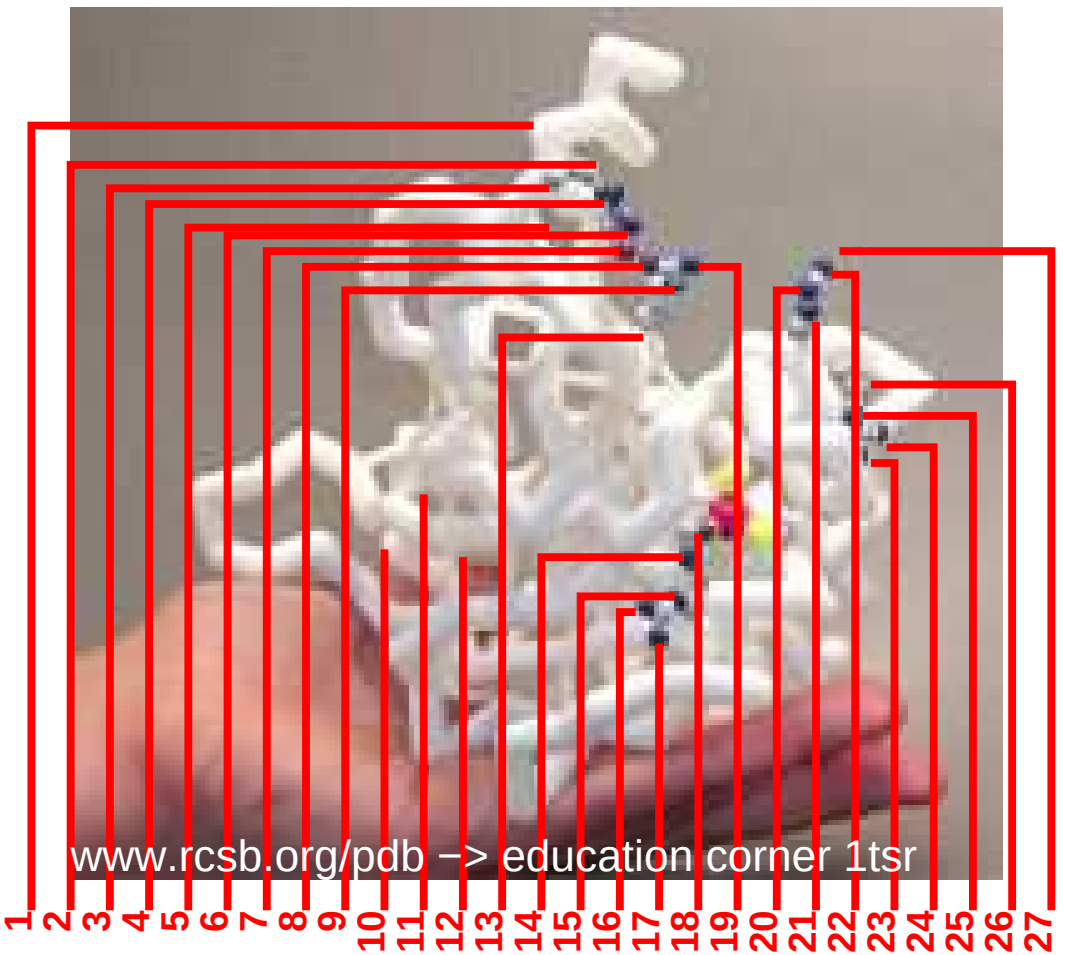
1. Motivation
2. Cell signalling modules (CSM)
3. P system framework Π_{CSM}
4. The KaiABC Oscillator: A circadian clock
5. Case Study Π_{KaiABC}
6. Network reconstruction by artificial evolution
7. The SBMLevolver:
   a two-level evolutionary algorithm
8. Evolved networks: a selection
9. Ongoing study: control system-based
   specification of circadian oscillators



Combinatorial Explosion of Protein Activation States

- Tumor suppressor protein p53: 27 phosphorylation sites
- Up to $2^{27} = 134,217,728$ distinguishable activation states
- Each state: individual constituent of reaction network



Cell Signalling Module

Characteristics

- Intracellular reaction network acting as functional unit
- Composed of proteins carrying phosphorylation sites
- Interactions between individual activation states

Facts

- Dynamical behaviour essential to understand function
- Often partially unknown
- Reconstruction as challenging task in systems biology
- Reverse engineering by integrative approach

Idea to manage complexity:
Capturing each protein by a specific string-object
instead of separate species per activation state

Specification of String-Objects

Assuming two alphabets: V (for protein names), V' (for protein properties); w.l.o.g $\#, \neg, * \notin V \cup V'$

Syntax for string-objects by regular set

$$S = V^+ \cdot \left(\{\#\} \cdot ((V')^+ \cup \{\neg\} \cdot (V')^+ \cup \{*\})\right)^*$$

Protein properties

- x: property x present (e.g. specific phosphate attached)
- $\neg x$: property x absent (e.g. specific phosphate removed)
- $*$: placeholder for arbitrary property setting

Examples

- $\mathrm{prot1\#p\#} * \mathrm{\#\neg p}$ (subsumes activation states of prot1)
- $\mathrm{KaiC\#\neg KaiA\#KaiB\#4}$ (prot. complex, 4 ligands attached)

$\Longrightarrow$ **Application of reaction rules requires string matching**

Π_{CSM}: System Components

Let $\langle S \rangle$ be the set of all multisets over S.

$$\Pi_{CSM} = (V, V', R_1, \ldots, R_r, \mathrm{f}_1, \ldots, \mathrm{f}_r, A, \mathrm{C}, \Delta\tau)$$

with

$R_i \in \langle S \rangle \times \langle S \rangle$ ..... is a reaction rule composed of two finite multisets

$\mathrm{f}_i : \langle S \rangle \longrightarrow \mathbb{N}$ ..... is a function corresponding to discrete kinetics of reaction R_i

$A \in \langle S \rangle$ ..... is a multiset of axioms representing the initial molecular configuration

$\mathrm{C} \in \mathbb{R}_+$ spatial capacity of the module (vessel or compartment)

$\Delta\tau \in \mathbb{R}_+$ ..... time discretisation interval

Π_{CSM}: Matching

Let S be a string-object syntax. Two string-objects match iff there is at least one common wild card-free representation:

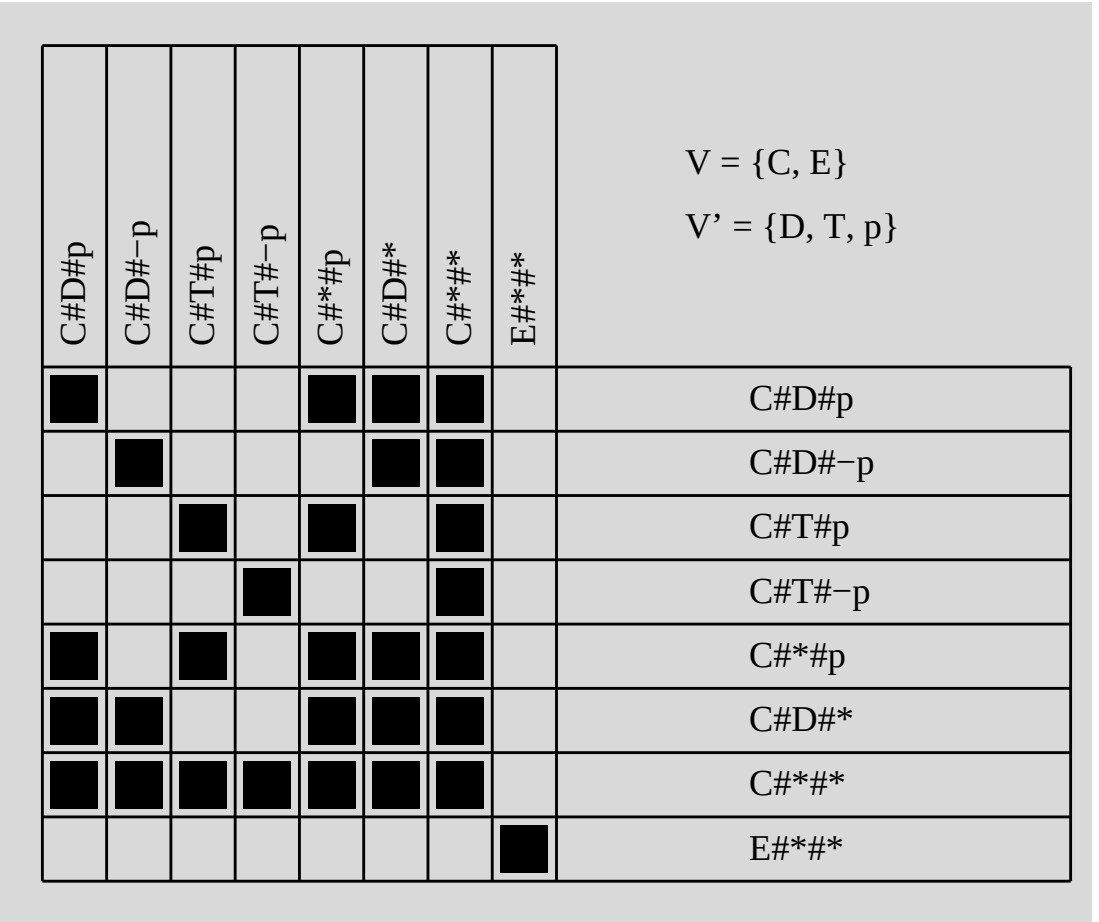
$$
\begin{aligned}
\mathit{Match} &\subseteq S \times S \\
\mathit{Match} &= \bigcup_{m \in \mathbb{N}} \{(p\#p_1\#p_2 \ldots \#p_m,\ s\#s_1\#s_2 \ldots \#s_m) \mid (p = s) \wedge \\
&\qquad \forall j \in \{1, \ldots, m\} : [(p_j = s_j) \vee (p_j = *) \vee (s_j = *) \vee \\
&\qquad\qquad ((p_j = \neg q) \wedge (s_j \neq q)) \vee ((s_j = \neg q) \wedge (p_j \neq q))]\}
\end{aligned}
$$

- *Match* is a symmetric relation
- Requires minimal similarity between string-objects with incomplete information
- Uncertainty interpreted as arbitrary replacement by available properties

Π_{CSM}: Matching

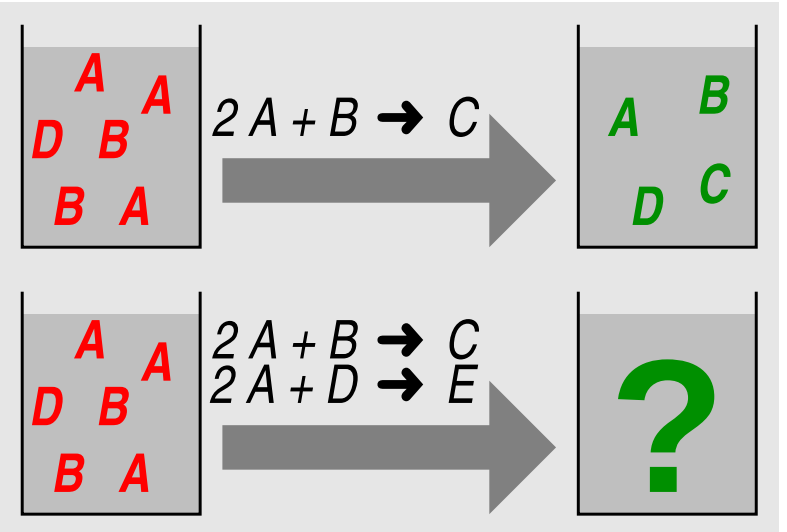
Let S be a string-object syntax. Two string-objects match iff there is at least one common wild card-free representation:

Example



Π_{CSM}: Dynamical System Behaviour (I)

- Successive progression of configuration $L_t \in \langle S \rangle$ over time $t \in \mathbb{N}$ starting from axioms A
- $\Delta\tau$: span between t and $t+1$
- Conflict handling by prioritisation of reaction rules



$$L_0 = L_{0,0} = A$$

$$L_{t,1} = \begin{cases} L_{t,0} \ominus Reactants_{t,1} \uplus Products_{t,1} & \text{if } Reactants_{t,1} \subseteq L_{t,0} \\ L_{t,0} & \text{otherwise} \end{cases}$$

$$\vdots$$

$$L_{t+1} = L_{t,r} = \begin{cases} L_{t,r-1} \ominus Reactants_{t,r} \uplus Products_{t,r} & \text{if } Reactants_{t,r} \subseteq L_{t,r-1} \\ L_{t,r-1} & \text{otherwise} \end{cases}$$

Π_{CSM}: Dynamical System Behaviour (II)

Estimation of multisets $Reactants_{t,j}$ and $Products_{t,j}$ at time t concerning reaction $R_j = (A_j, B_j) \in \langle S \rangle \times \langle S \rangle$ denoted

$$A_j(a_1)\ a_1 + \ldots + A_j(a_p)\ a_p \longrightarrow B_j(b_1)\ b_1 + \ldots + B_j(b_q)\ b_q$$

includes

- Matching between string-objects in L_t and those in A_j
- Consideration of stoichiometry captured by multisets A_j, B_j
- Evaluation of kinetic law expressed by scalar function f_j

$$Reactants_{t,j} = \biguplus_{e_1 \in Match(a_1)} \ldots \biguplus_{e_p \in Match(a_p)} f_j\left(\{(e_1,\infty),\ldots,(e_p,\infty)\} \cap L_{t,j-1}\right) \cdot \left\{(e_1, A_j(a_1)),\ldots,(e_p, A_j(a_p))\right\}$$

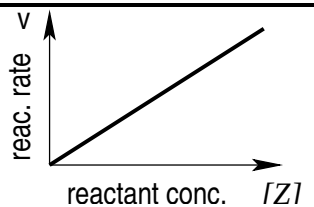
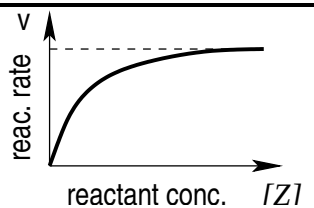
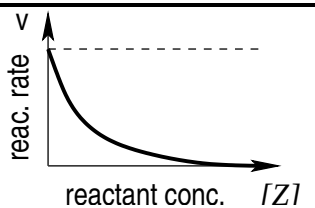
$$Products_{t,j} = \biguplus_{e_1 \in Match(a_1)} \ldots \biguplus_{e_p \in Match(a_p)} f_j\left(\{(e_1,\infty),\ldots,(e_p,\infty)\} \cap L_{t,j-1}\right) \cdot \left\{(b_1, B_j(b_1)),\ldots,(b_q, B_j(b_q))\right\}$$

Π_{CSM}: Discrete Reaction Kinetics

Scalar function f_j provides number of turns for application of reaction rule R_j. Rate constant: $k_j = \hat{k}_j \cdot \mathrm{C} \cdot \Delta\tau$ (Euler).

$$\mathrm{f}_j(L_t) = \left\lfloor k_j \prod_{\forall \alpha \in Match(A_j) \cap Match(L_t) : (R_j = (A_j, B_j))} \hat{\mathrm{f}}(L_t(\alpha))^{|Match(A_j) \cap \{(\alpha, \infty)\}|} \right\rfloor$$

Selected kinetic laws $\hat{\mathrm{f}}([Z])$

Kinetics	Activation	Repression
Mass-Action (no saturation)	 $\hat{\mathrm{f}}([Z]) = [Z]$	–
Michaelis-Menten (saturation)	 $\hat{\mathrm{f}}([Z]) = \frac{[Z]}{\Theta + [Z]}$	 $\hat{\mathrm{f}}([Z]) = \left(1 - \frac{[Z]}{\Theta + [Z]}\right)$

Circadian Clocks

Characteristics

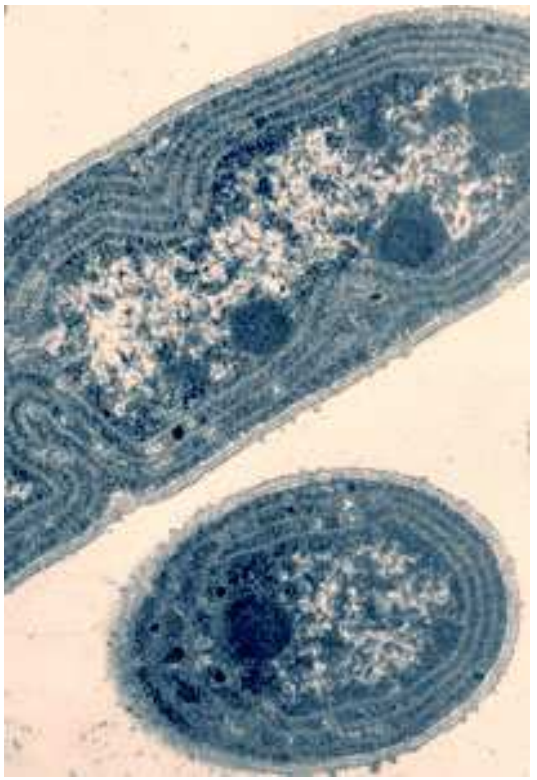
- Self-sustained biochemical oscillators
- Period of approx. 24 hours persisting under constant environmental conditions (e.g. constant darkness)
- Temperature compensation within physiological range
- Capability of entrainment by external stimuli (e.g. light/dark or temperature cycles)
- Reaction system with at least one feedback loop

High scientific impact because . . .

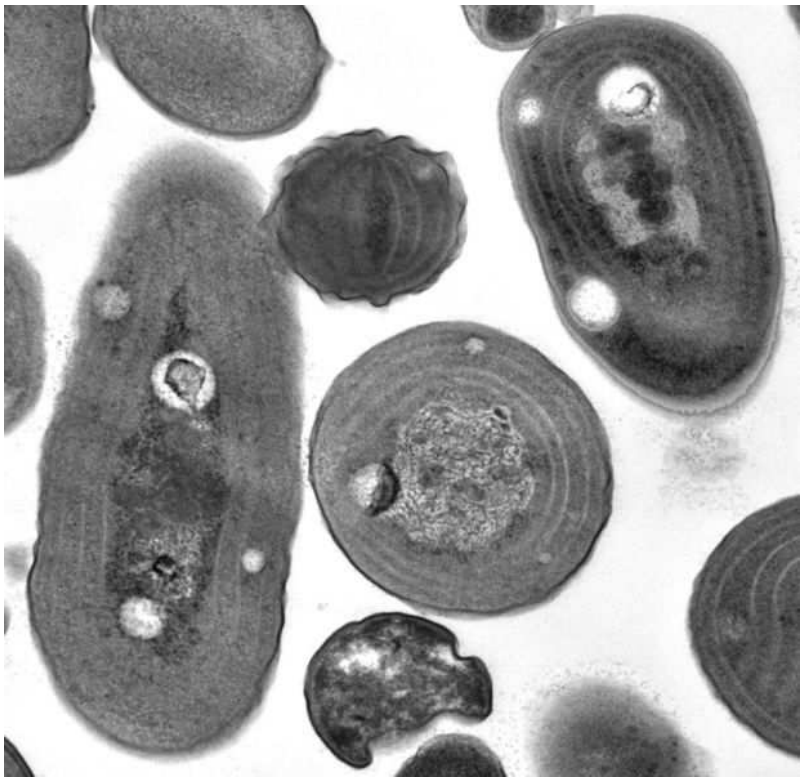
- Circadian clock as a potential universal property of life
- Self-sustainability and high precision of bio-oscillators
- Chronobiological control systems for manifold processes
- Several independent evolutionary origins assumed

Cyanobacterium *Synechococcus elongatus*

"Simplest cells known to exhibit circadian phenomena"



www.genome.jgi–psf.org



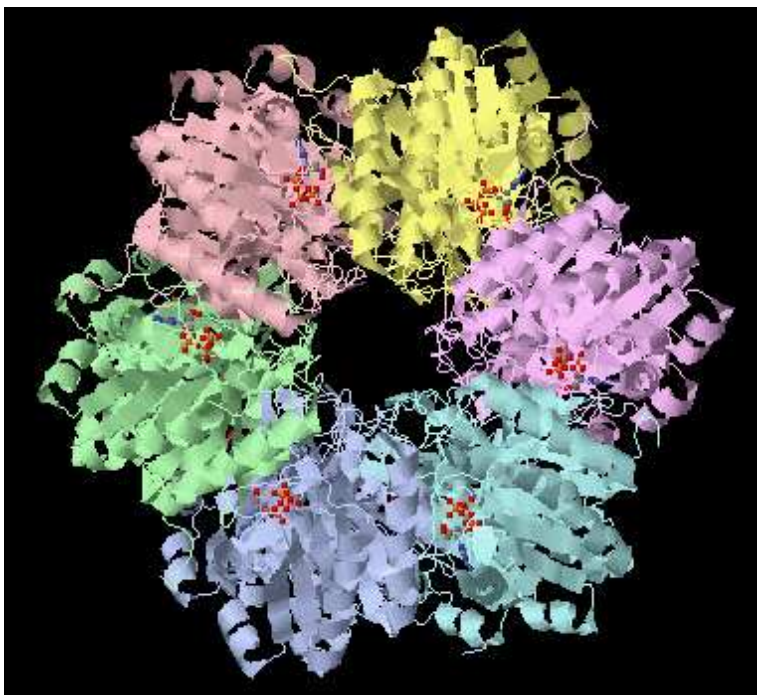
www.wikipedia.org

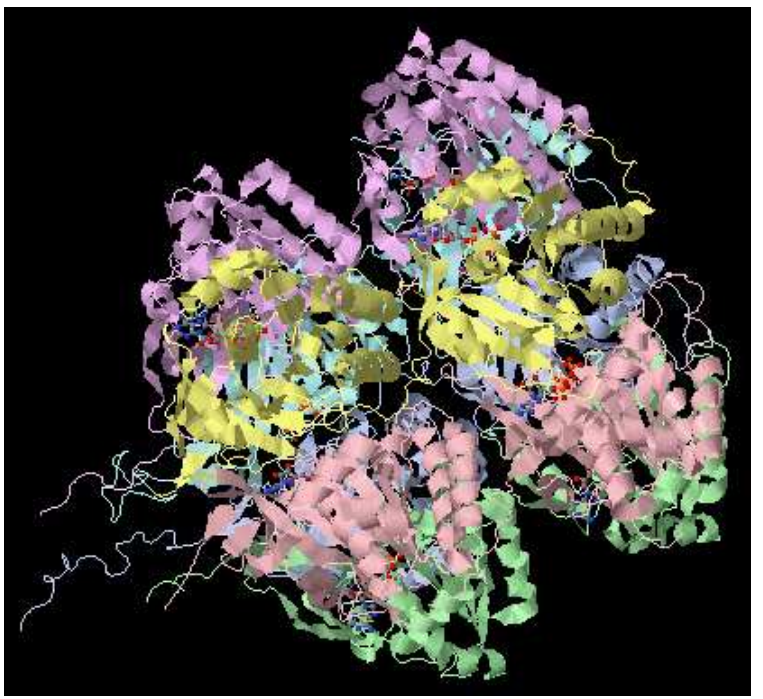
Prokaryotic autotrophic picoplankton in tropical oceans

Genome: 2.4 . . . 2.7 Mbp

Components of Circadian Clock: Key Protein KaiC

- Homohexamer (“double doughnut”) with 12 ATP molecules
- Protein kinase (transferase), length: 519 residues

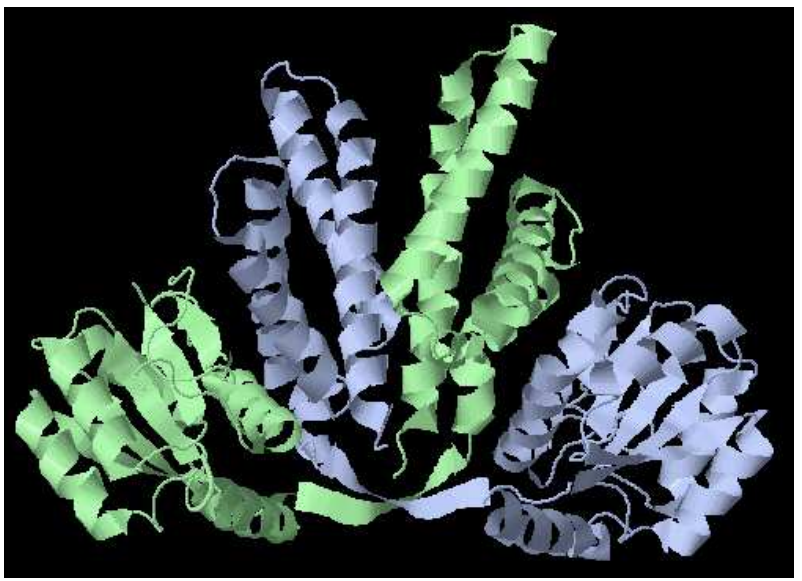




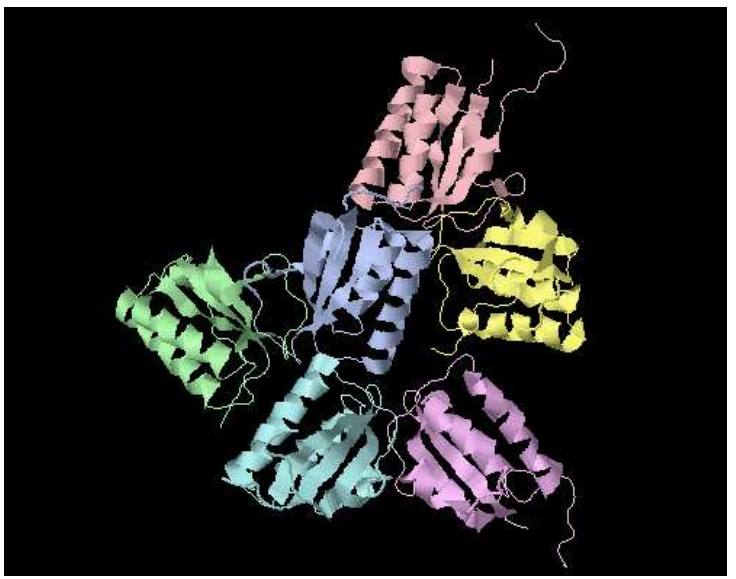
PDB Protein Data Bank, ID: 2gbl, www.rcsb.org/pdb

Key Clock Proteins KaiA and KaiB

- KaiA: protein binding molecular function reported, length: 289 residues
- KaiB: no further molecular function reported, length: 108 residues

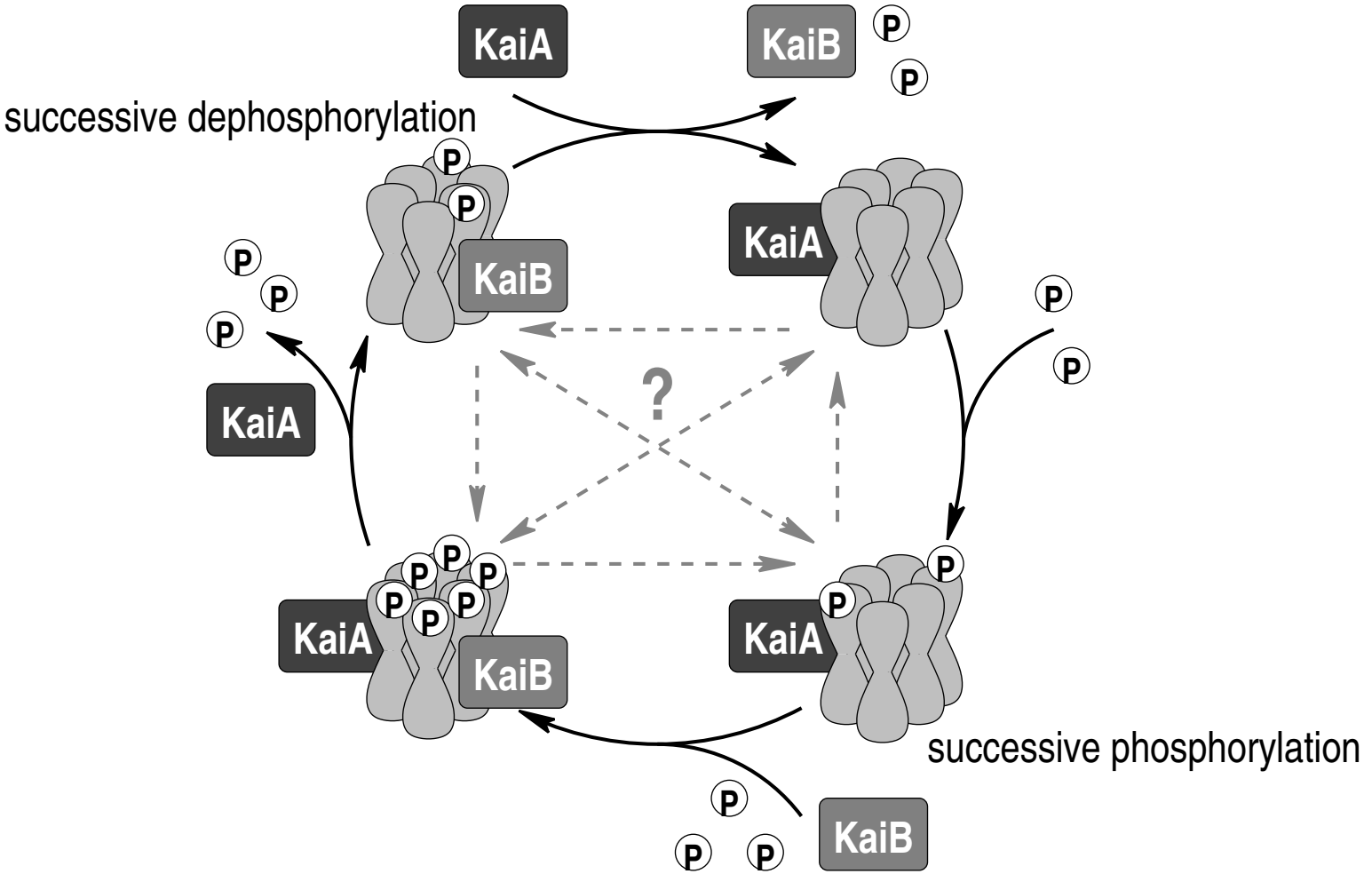


KaiA protein structure from PDB Protein Data Bank, ID: 1r8j
www.rcsb.org/pdb



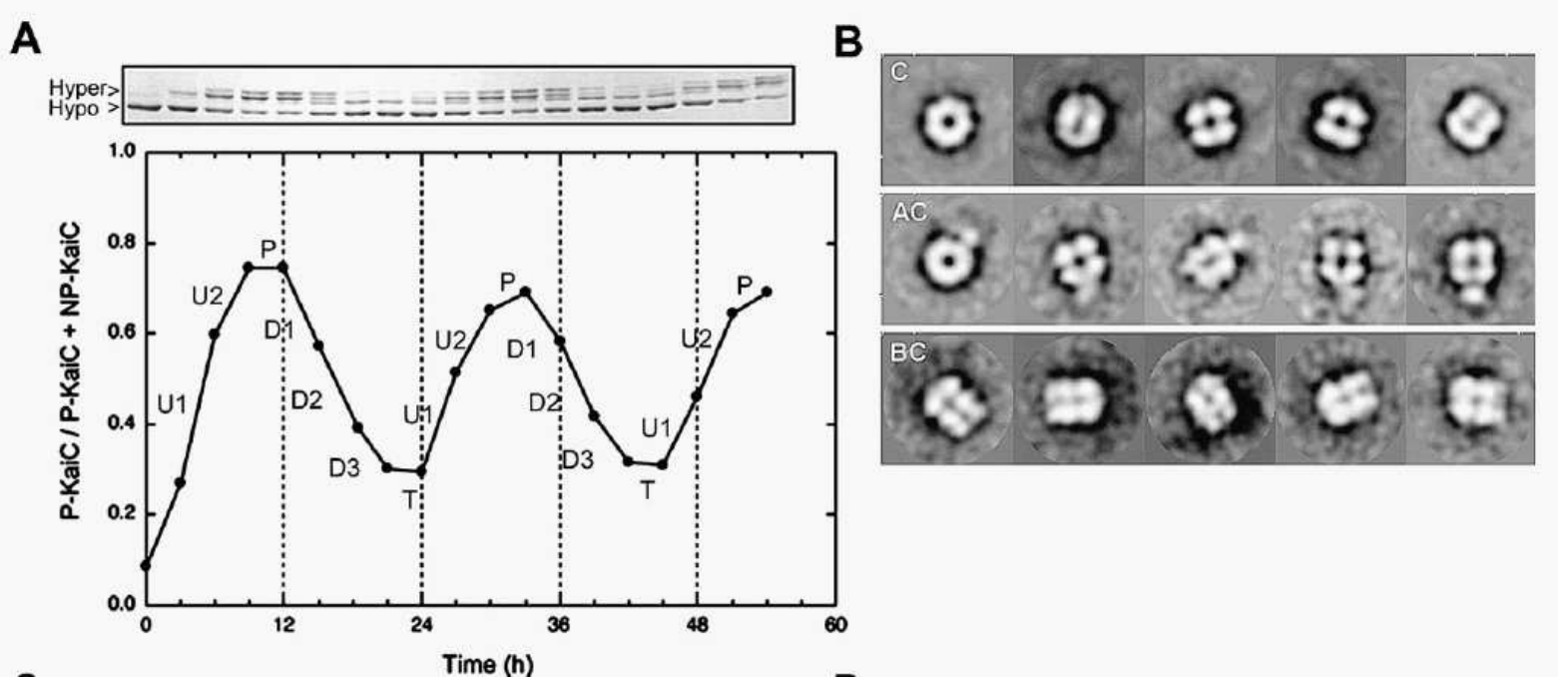
KaiB protein structure from PDB Protein Data Bank, ID: 2qke

KaiABC Oscillator: Reaction Cycle



Incomplete information about interphase feedback loops

KaiC Oscillating Behaviour in Seven Phases



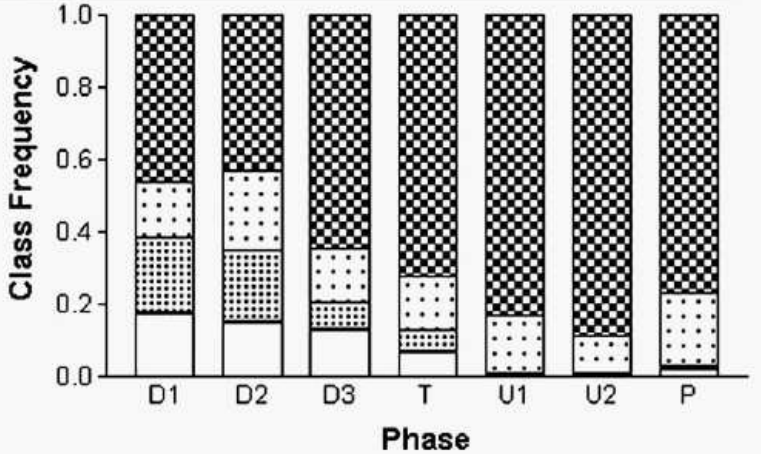
(A) PAGE gel and oscillation phases: U1, U2 (upward), P (peak), D1, D2, D3 (descent), T (trough)
(B) Representative electron microscopic images of KaiC (C) and complexes KaiA●KaiC (AC), KaiB●KaiC (BC)

T. Mori, D.R. Williams, M.O. Byrne, X. Qin, M. Egli, H.S. Mchaourab, P.L. Stewart, C.H. Johnson. Elucidating the Ticking of an In Vitro Circadian Clockwork. PLoS Biology **5(4)**:841–853, 2007, doi: 10.1371/journal.pbio.0050093

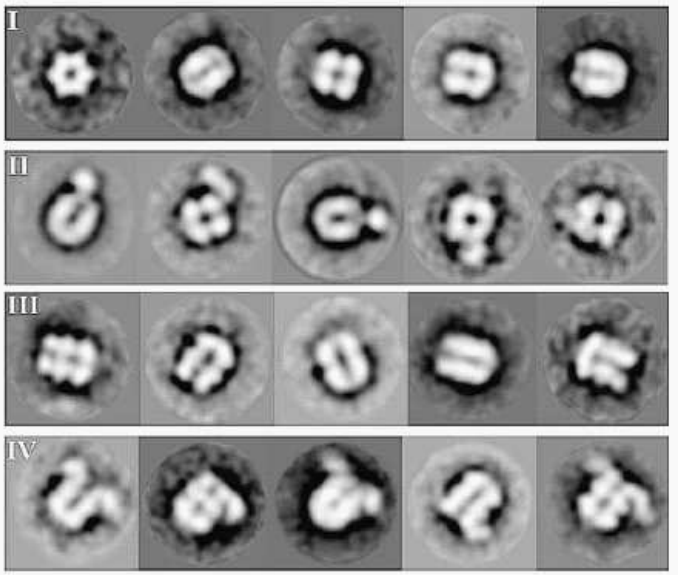
Relative Frequencies of KaiC and Complexes in Phases D1, D2, D3, T, U1, U2, P

C

	D1	D2	D3	T	U1	U2	P
I	0.43	0.40	0.52	0.69	0.83	0.90	0.76
II	0.16	0.23	0.20	0.15	0.16	0.10	0.20
III	0.22	0.20	0.12	0.07	0.00	0.00	0.01
IV	0.19	0.18	0.16	0.09	0.01	0.01	0.04



D



(C) Assignment of frequency classes
I KaiC hexamers alone, **II** KaiA●KaiC, **III** KaiB●KaiC, **IV** KaiA●KaiB●KaiC

(D) Representative electron microscopic images of classes

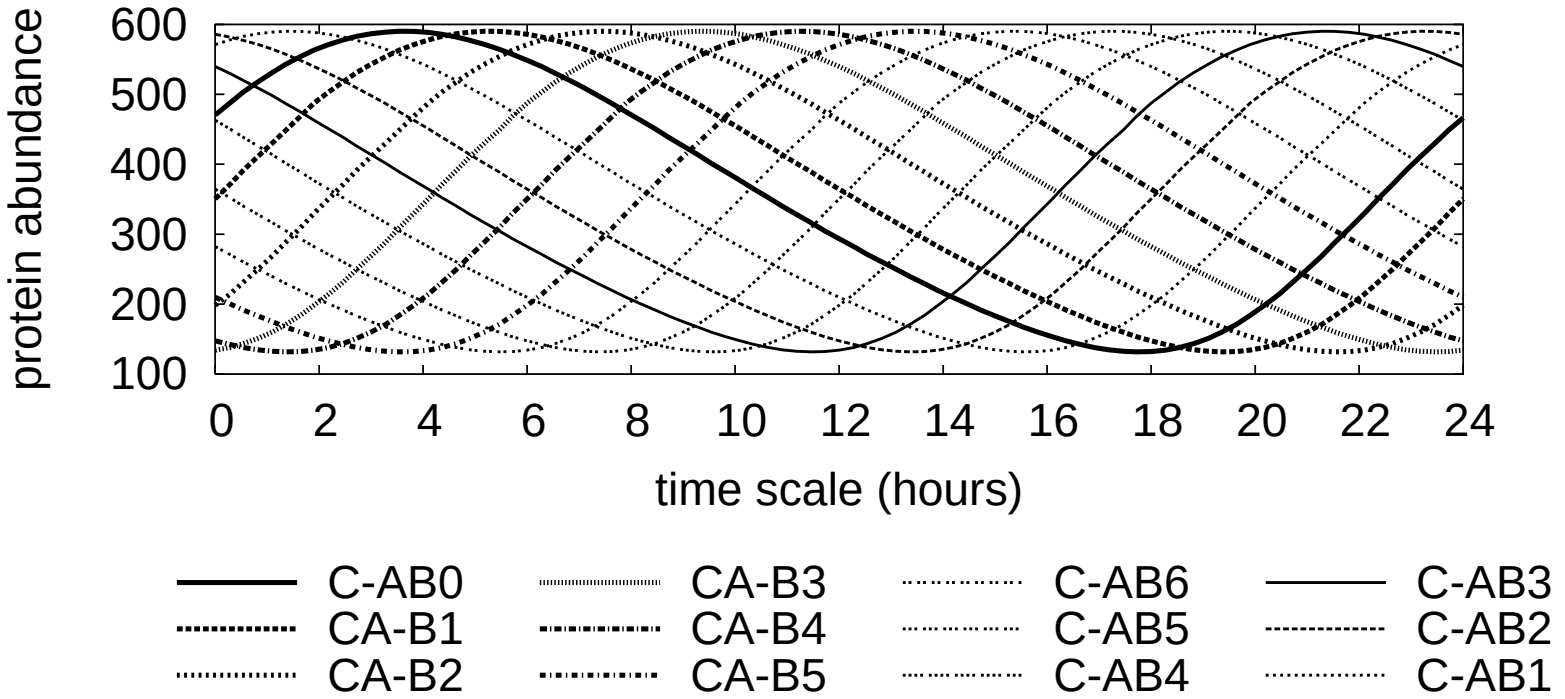
T. Mori, D.R. Williams, M.O. Byrne, X. Qin, M. Egli, H.S. Mchaourab, P.L. Stewart, C.H. Johnson. Elucidating the Ticking of an In Vitro Circadian Clockwork. PLoS Biology **5(4)**:841–853, 2007, doi: 10.1371/journal.pbio.0050093

The Model Π_{KaiABC} at a Glance

$$
\begin{aligned}
\Pi_{KaiABC} &= (V, V', R_1, \ldots, R_{17}, \mathrm{f}_1, \ldots, \mathrm{f}_{17}, A, \mathrm{C}, \Delta\tau) \\
V &= \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \ldots\ldots\ldots\ldots\ldots\ldots \text{identifiers of proteins KaiA, KaiB, KaiC} \\
V' &= \{\mathrm{A}, \mathrm{B}\} \cup \ldots\ldots\ldots\ldots\ldots \text{KaiA, KaiB within a complex associated to KaiC} \\
&\quad \{0, 1, 2, 3, 4, 5, 6\} \ldots\ldots \text{number of attached phosphates} \\
R_1 &= \mathrm{C}\#\neg\mathrm{A}\#\mathrm{B}\#0 + \mathrm{A} \longrightarrow \mathrm{C}\#\mathrm{A}\#\neg\mathrm{B}\#1 + \mathrm{B} \\
R_2 &= \mathrm{C}\#\mathrm{A}\#*\#1 + \mathrm{A} \longrightarrow \mathrm{C}\#\mathrm{A}\#*\#2 + \mathrm{A} \\
R_3 &= \mathrm{C}\#\mathrm{A}\#*\#2 + \mathrm{A} \longrightarrow \mathrm{C}\#\mathrm{A}\#*\#3 + \mathrm{A} \\
R_4 &= \mathrm{C}\#\mathrm{A}\#*\#3 + \mathrm{A} \longrightarrow \mathrm{C}\#\mathrm{A}\#*\#4 + \mathrm{A} \\
R_5 &= \mathrm{C}\#\mathrm{A}\#*\#4 + \mathrm{A} \longrightarrow \mathrm{C}\#\mathrm{A}\#*\#5 + \mathrm{A} \\
R_6 &= \mathrm{C}\#\mathrm{A}\#\neg\mathrm{B}\#5 + \mathrm{B} \longrightarrow \mathrm{C}\#\neg\mathrm{A}\#\mathrm{B}\#6 + \mathrm{A} \\
R_7 &= \mathrm{C}\#*\#\mathrm{B}\#6 + \mathrm{B} \longrightarrow \mathrm{C}\#*\#\mathrm{B}\#5 + \mathrm{B} \\
R_8 &= \mathrm{C}\#*\#\mathrm{B}\#5 + \mathrm{B} \longrightarrow \mathrm{C}\#*\#\mathrm{B}\#4 + \mathrm{B} \\
R_9 &= \mathrm{C}\#*\#\mathrm{B}\#4 + \mathrm{B} \longrightarrow \mathrm{C}\#*\#\mathrm{B}\#3 + \mathrm{B} \\
R_{10} &= \mathrm{C}\#*\#\mathrm{B}\#3 + \mathrm{B} \longrightarrow \mathrm{C}\#*\#\mathrm{B}\#2 + \mathrm{B} \\
R_{11} &= \mathrm{C}\#*\#\mathrm{B}\#2 + \mathrm{B} \longrightarrow \mathrm{C}\#*\#\mathrm{B}\#1 + \mathrm{B} \\
R_{12} &= \mathrm{C}\#*\#\mathrm{B}\#1 + \mathrm{B} \longrightarrow \mathrm{C}\#*\#\mathrm{B}\#0 + \mathrm{B} \\
R_{13} &= \mathrm{C}\#\neg\mathrm{A}\#\mathrm{B}\#* + \mathrm{A} \longrightarrow \mathrm{C}\#\mathrm{A}\#\neg\mathrm{B}\#* + \mathrm{B} \\
R_{14} &= \mathrm{C}\#\mathrm{A}\#\neg\mathrm{B}\#* + \mathrm{B} \longrightarrow \mathrm{C}\#\neg\mathrm{A}\#\mathrm{B}\#* + \mathrm{A} \\
R_{15} &= \mathrm{A} \longrightarrow \emptyset \\
R_{16} &= \mathrm{B} \longrightarrow \emptyset \\
R_{17} &= \mathrm{C}\#*\#*\#* \longrightarrow \emptyset \\
&\ldots
\end{aligned}
$$

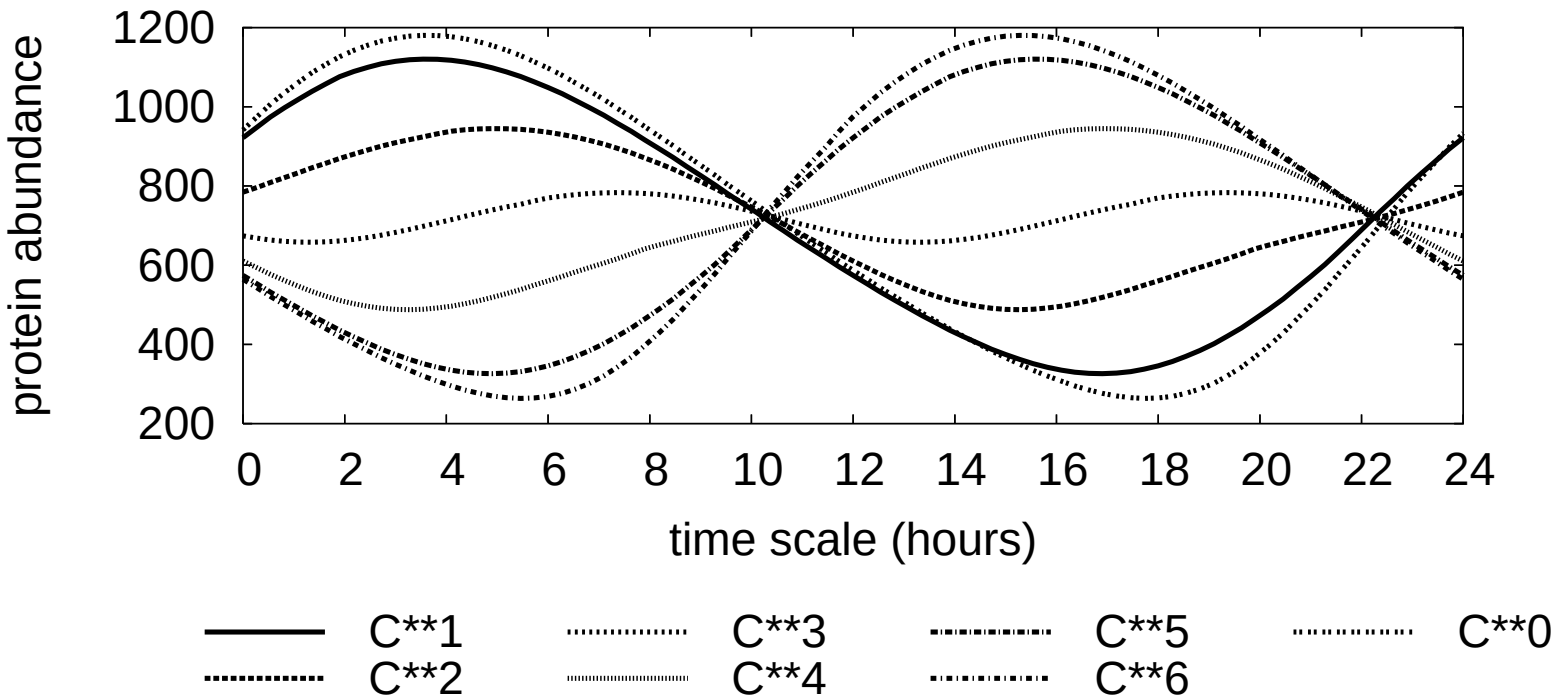
Discrete Michaelis-Menten kinetics

Simulation Results: Individual KaiABC Subproducts



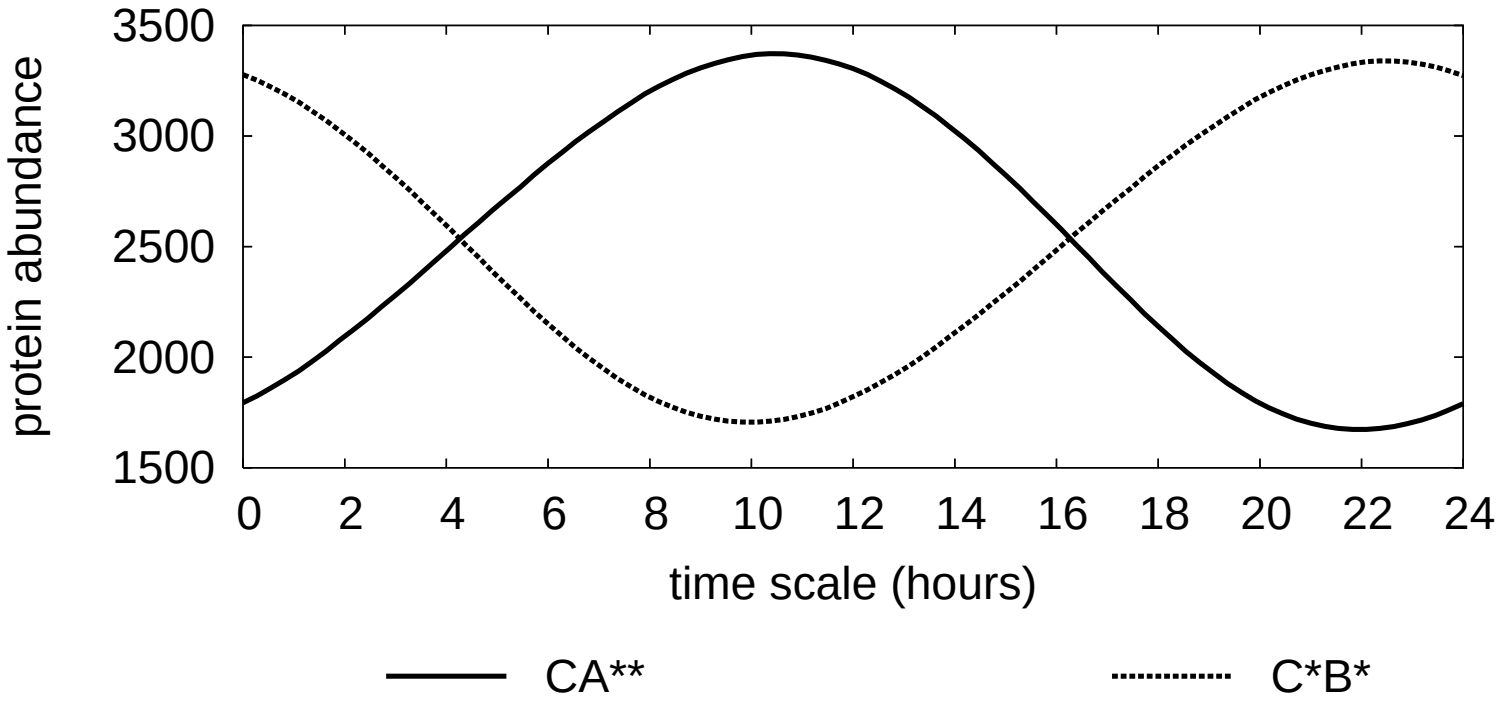
Temporal courses of 12 specific KaiABC subproducts representing the process status of the reaction cycle. Kinetic parameters and initial amounts adjusted in a way to obtain a period of $\approx$ 24 hours and symmetry among individual oscillations.

Focussing on the Level of Phosphorylation



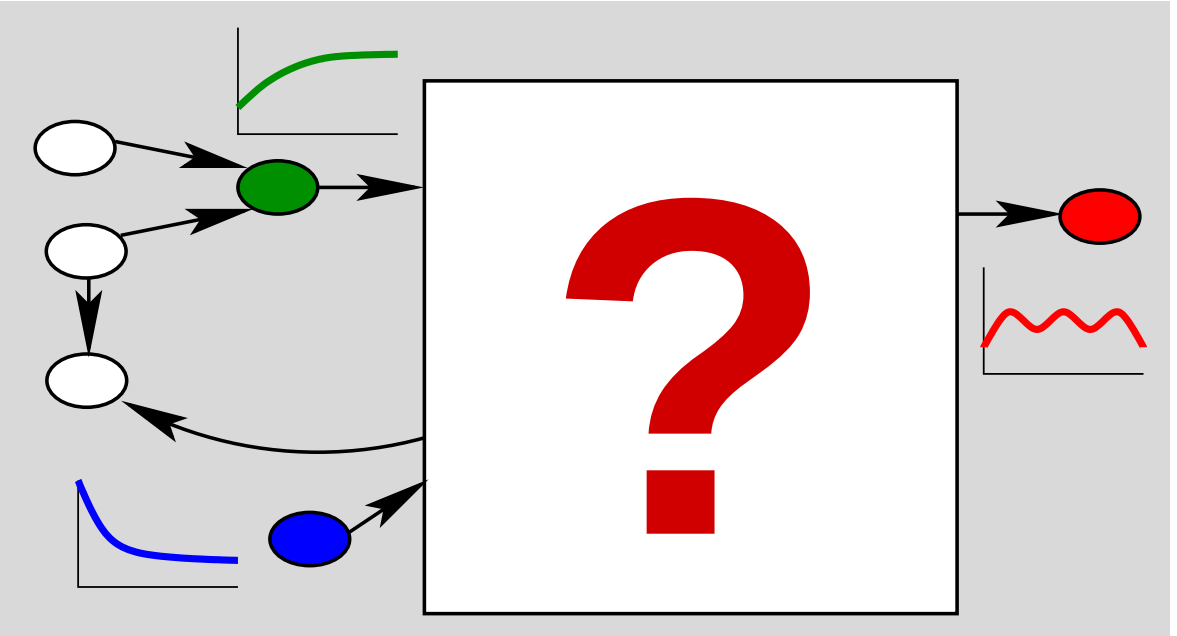
Temporal courses of KaiABC subproducts subsumed by their level of phosphorylation ranging from 0 to 6. Kinetic parameters and initial amounts adjusted in a way to obtain a period of $\approx$ 24 hours and symmetry among individual oscillations.

Focussing on KaiABC Complex Formation



Temporal courses of KaiABC subproducts separated into two groups by association of KaiA resp. KaiB to KaiC. Kinetic parameters and initial amounts adjusted in a way to obtain a period of $\approx$ 24 hours and symmetry among individual oscillations.

Reaction Network Reconstruction from Scratch



- Partially unknown topology
- Some behavioural data available
- Reconstruction of appropriate reaction network candidates
- Capturing ideas and inspirations for network topologies and parameterisation suitable for specific task

Reaction Network Reconstruction: A Challenging Task

Exhaustive candidate enumeration

- Exponential growth of search space:
  n species $\longrightarrow$ 2^{2n} possible first-order reactions

Finding homologies

- Employ synergetic effects: known networks with similar functionality could be adapted

Bottom-up engineering

- Provide small functional units and combine them towards entire network (constructive approach)

Learning strategies

- Reduce a huge full network by successive weighting of reactions along with available behavioural data

Artificial network evolution

- Universal heuristics adopted from biological evolution

Reaction Network Reconstruction: A Challenging Task

Exhaustive candidate enumeration

- Exponential growth of search space:
  n species $\longrightarrow$ 2^{2n} possible first-order reactions

Finding homologies

- Employ synergetic effects: known networks with similar functionality could be adapted

Bottom-up engineering

- Provide small functional units and combine them towards entire network (constructive approach)

Learning strategies

- Reduce a huge full network by successive weighting of reactions along with available behavioural data

Artificial network evolution

- Universal heuristics adopted from biological evolution

Reaction Network Reconstruction: A Challenging Task

Exhaustive candidate enumeration

- Exponential growth of search space:
  n species $\longrightarrow$ 2^{2n} possible first-order reactions

Finding homologies

- Employ synergetic effects: known networks with similar functionality could be adapted

Bottom-up engineering

- Provide small functional units and combine them towards entire network (constructive approach)

Learning strategies

- Reduce a huge full network by successive weighting of reactions along with available behavioural data

Artificial network evolution

- Universal heuristics adopted from biological evolution

Reaction Network Reconstruction: A Challenging Task

Exhaustive candidate enumeration

- Exponential growth of search space:
  n species $\longrightarrow$ 2^{2n} possible first-order reactions

Finding homologies

- Employ synergetic effects: known networks with similar functionality could be adapted

Bottom-up engineering

- Provide small functional units and combine them towards entire network (constructive approach)

Learning strategies

- Reduce a huge full network by successive weighting of reactions along with available behavioural data

Artificial network evolution

- Universal heuristics adopted from biological evolution

Reaction Network Reconstruction: A Challenging Task

Exhaustive candidate enumeration

- Exponential growth of search space:
  n species $\longrightarrow$ 2^{2n} possible first-order reactions

Finding homologies

- Employ synergetic effects: known networks with similar functionality could be adapted

Bottom-up engineering

- Provide small functional units and combine them towards entire network (constructive approach)

Learning strategies

- Reduce a huge full network by successive weighting of reactions along with available behavioural data

Artificial network evolution

- Universal heuristics adopted from biological evolution

Why Artificial Evolution for Network Reverse Engineering

- Systems Biology deals with interplay of biological components rather than components themselves.

www.wordpress.com

- Accumulation of small modifications in component's interplay can result in a new quality of the entire network.
  $\Longrightarrow$ Artificial evolution can explore network structure.
- Help in understanding emergence of biological complexity.
  $\Longrightarrow$ Evolution becomes observable.
- Furthermore, bio-inspired approaches provide a flexible, fault-tolerant, reliable paradigm.
  $\Longrightarrow$ Artificial evolution can find unexpected, unconventional solutions.

Why Artificial Evolution for Network Reverse Engineering

www.wordpress.com

- Systems Biology deals with interplay of biological components rather than components themselves.
- Accumulation of small modifications in component's interplay can result in a new quality of the entire network.

  $\Longrightarrow$ Artificial evolution can explore network structure.
- Help in understanding emergence of biological complexity.

  $\Longrightarrow$ Evolution becomes observable.
- Furthermore, bio-inspired approaches provide a flexible, fault-tolerant, reliable paradigm.

  $\Longrightarrow$ Artificial evolution can find unexpected, unconventional solutions.

Why Artificial Evolution for Network Reverse Engineering

www.wordpress.com

- Systems Biology deals with interplay of biological components rather than components themselves.
- Accumulation of small modifications in component's interplay can result in a new quality of the entire network.
  $\Longrightarrow$ Artificial evolution can explore network structure.
- Help in understanding emergence of biological complexity.
  $\Longrightarrow$ Evolution becomes observable.
- Furthermore, bio-inspired approaches provide a flexible, fault-tolerant, reliable paradigm.
  $\Longrightarrow$ Artificial evolution can find unexpected, unconventional solutions.

Why Artificial Evolution for Network Reverse Engineering

www.wordpress.com

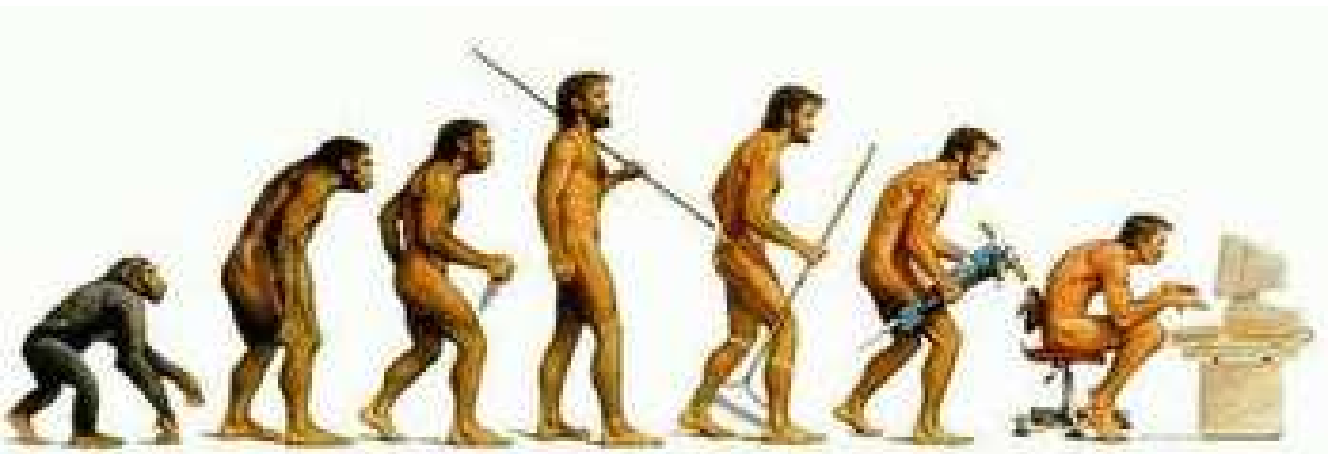
- Systems Biology deals with interplay of biological components rather than components themselves.
- Accumulation of small modifications in component's interplay can result in a new quality of the entire network.
  $\Longrightarrow$ Artificial evolution can explore network structure.
- Help in understanding emergence of biological complexity.
  $\Longrightarrow$ Evolution becomes observable.
- Furthermore, bio-inspired approaches provide a flexible, fault-tolerant, reliable paradigm.
  $\Longrightarrow$ Artificial evolution can find unexpected, unconventional solutions.

Evolutionary Computing

- Abstraction and formalisation of evolutionary processes
- Individuals (genotype, phenotype) and population
- Evolutionary operators along with fitness evaluation
- Heuristical optimisation technique, experimentally driven

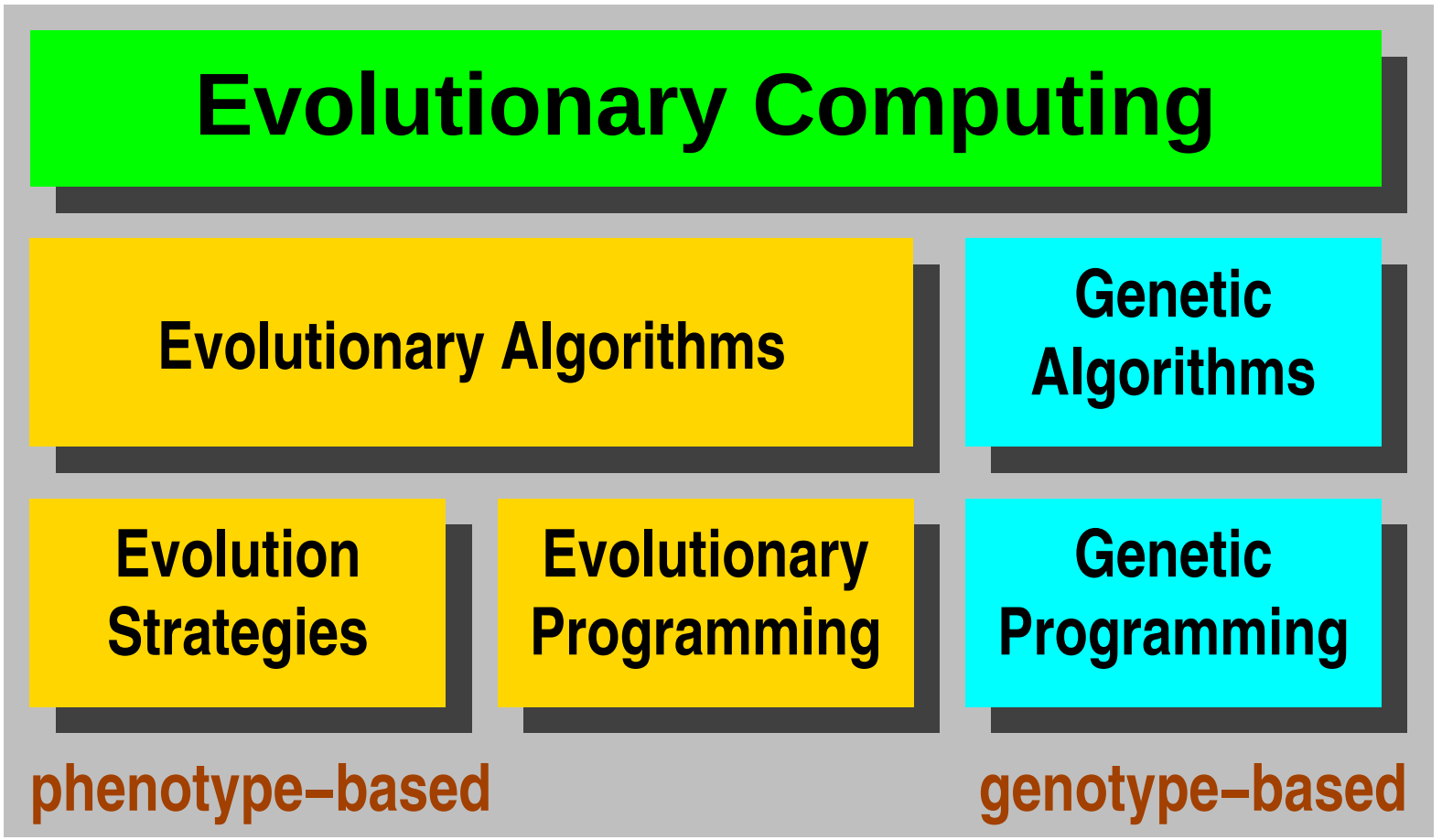
Artificial evolution

- Initiated by Friedmann 1956
- Pioneers: Rechenberg, Schwefel, Fogel, Holland, Banzhaf, Koza, Sauro, . . .

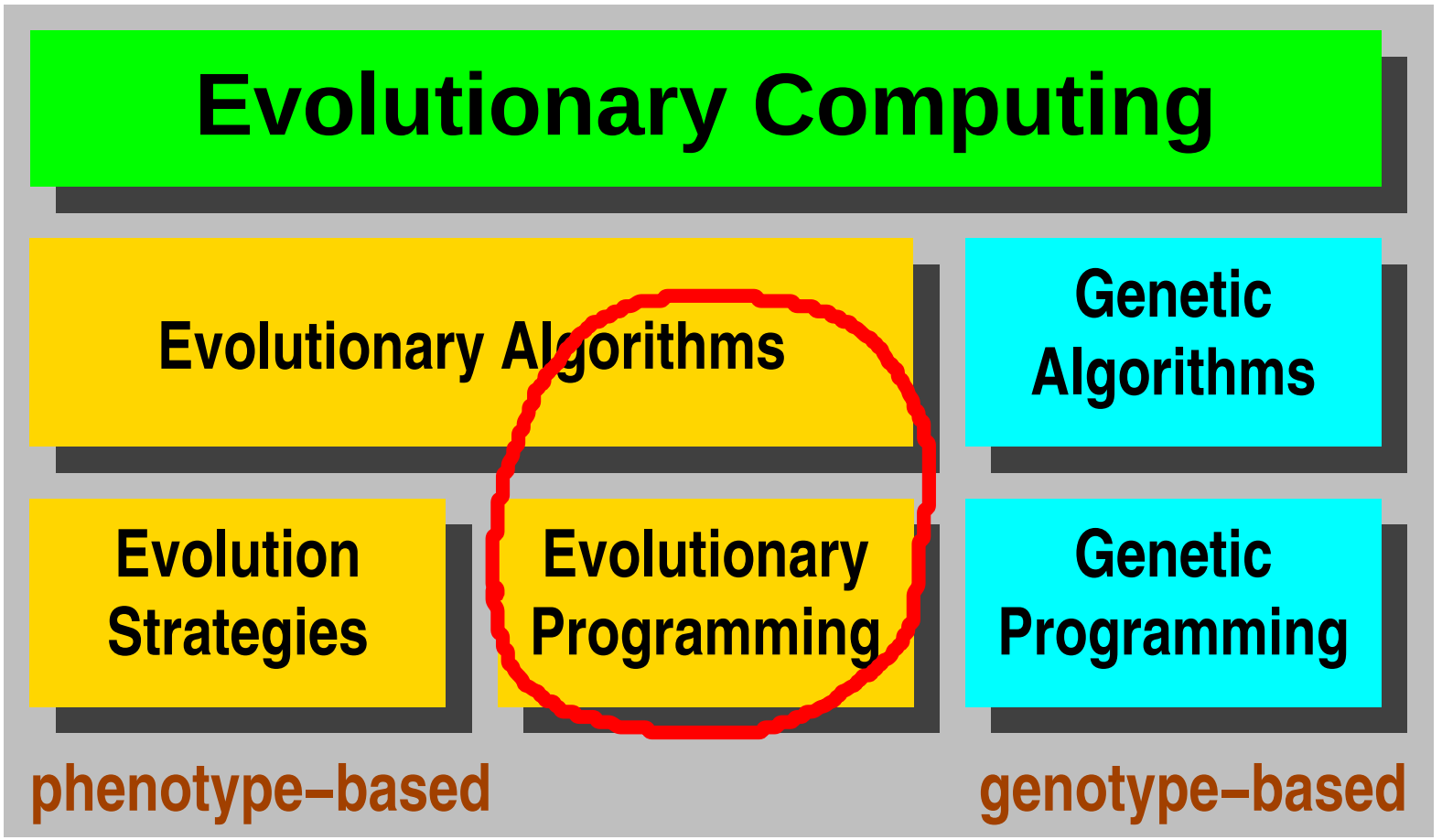


pics.goingon.com

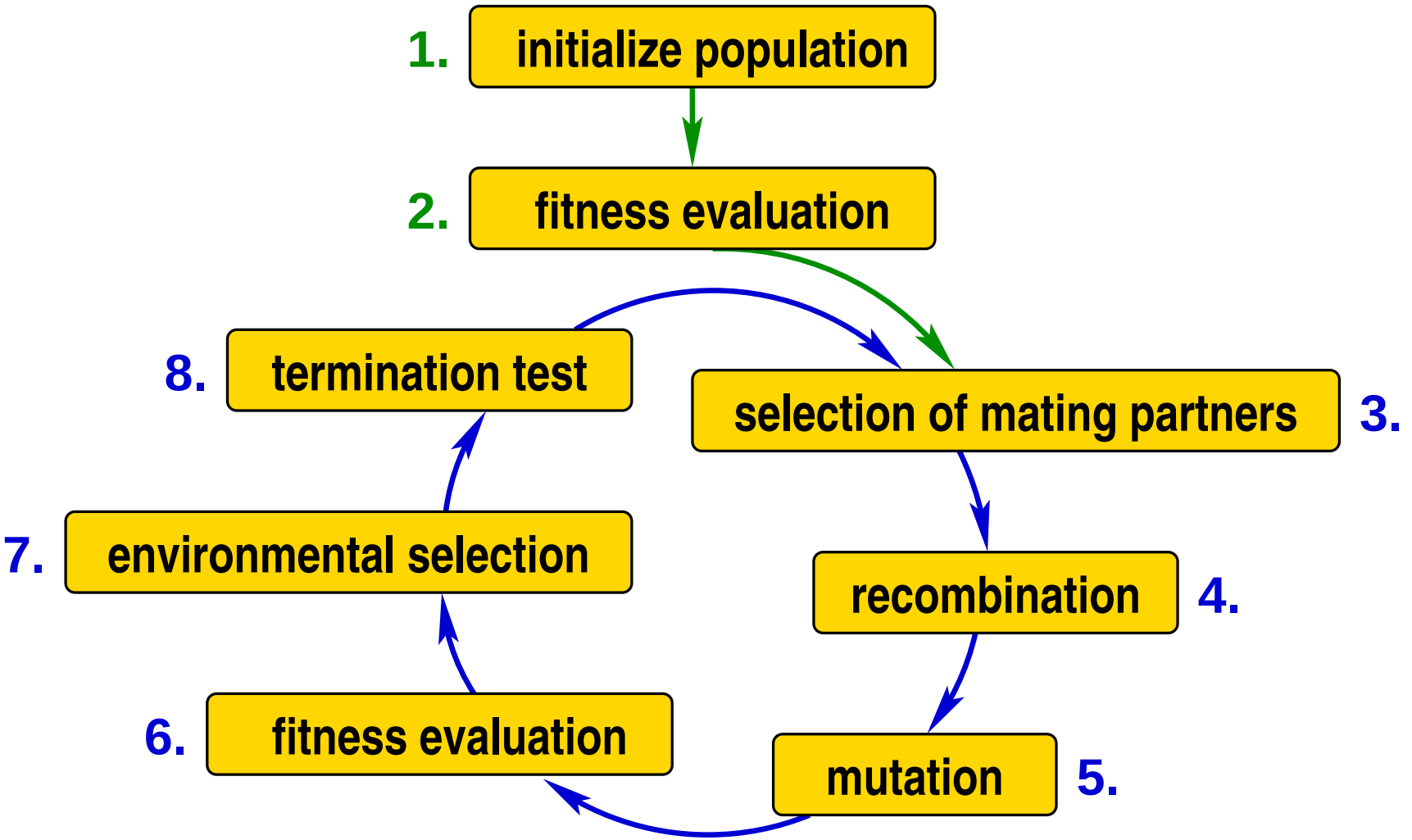
Facets and Specialties



Facets and Specialties

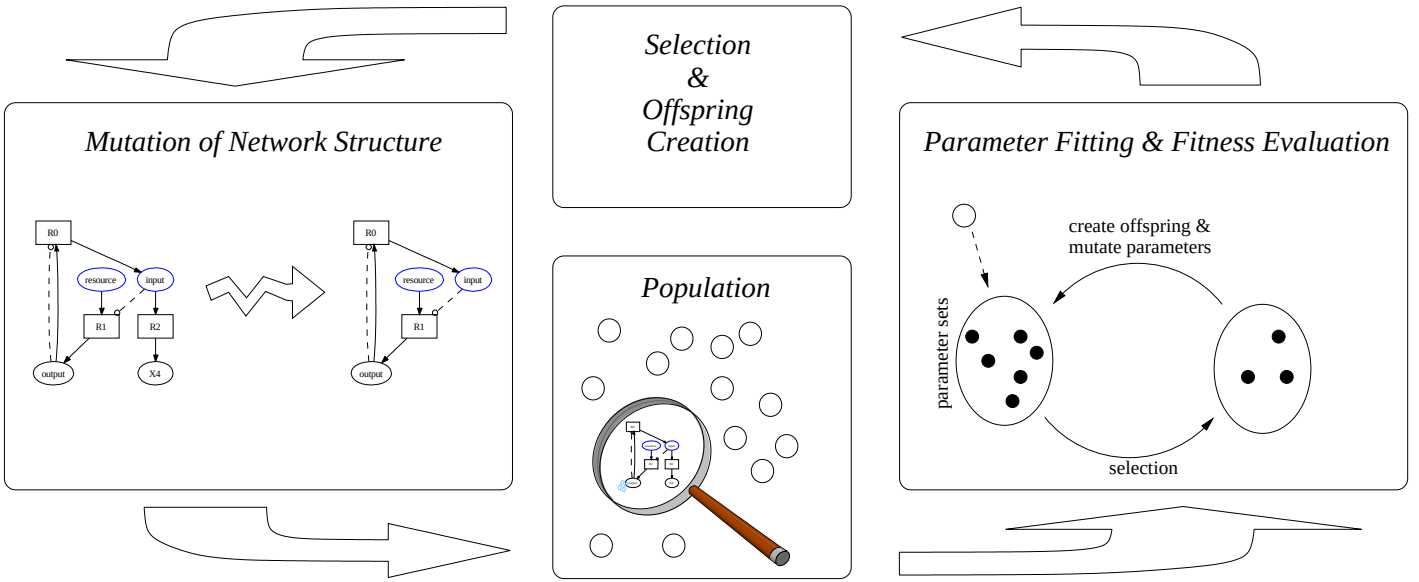


Central Loop in Evolutionary Algorithms



SBMLevolver: Two-Level Evolutionary Algorithm

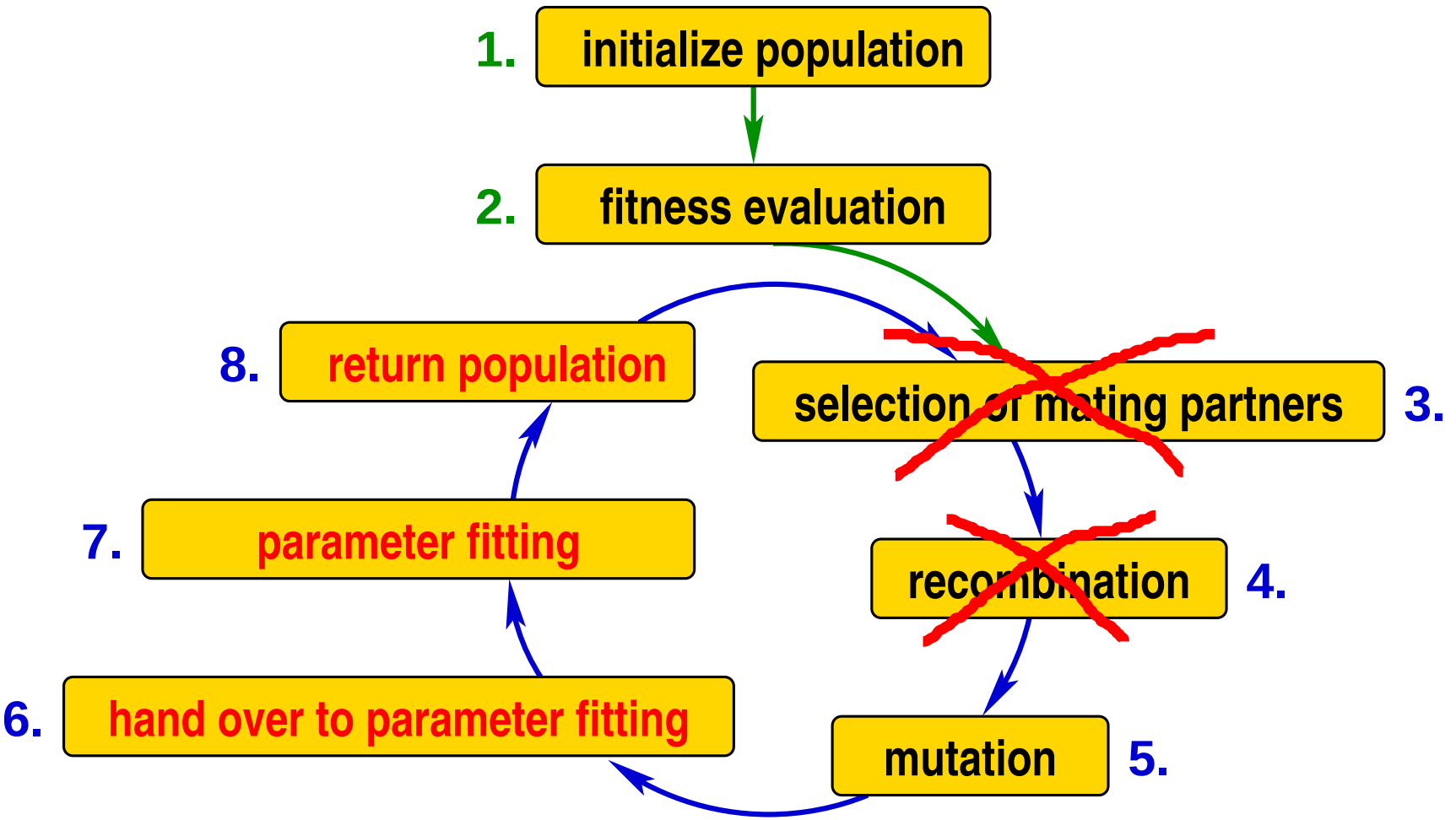
- Separation of structural evolution from parameter fitting
- Idea: parameters can adapt to mutated network structure



- Upper level: network structure
- Lower level: kinetic parameter fitting

$\Longrightarrow$ open-source freeware:
http://users.minet.uni-jena.de/∼biosys/esignet

Structural Evolution



Initialization of Network Population

Initial population configurable,
typically 50 . . . 100 network individuals as SBML files

Empty

- Network reconstruction from scratch

Randomly choosen

- Individual networks randomly chosen, upper/lower limits for numbers of species, reactions, and kinetic parameter values

Taken from imported SBML file

- Generate a number of file copies
- Dedicated species, reactions, and kinetic parameters can be marked as fixed during evolution

Initialization of Network Population

Initial population configurable,
typically 50 . . . 100 network individuals as SBML files

Empty

- Network reconstruction from scratch

Randomly choosen

- Individual networks randomly chosen, upper/lower limits for numbers of species, reactions, and kinetic parameter values

Taken from imported SBML file

- Generate a number of file copies
- Dedicated species, reactions, and kinetic parameters can be marked as fixed during evolution

Initialization of Network Population

Initial population configurable,
typically 50 . . . 100 network individuals as SBML files

Empty

- Network reconstruction from scratch

Randomly choosen

- Individual networks randomly chosen, upper/lower limits for numbers of species, reactions, and kinetic parameter values

Taken from imported SBML file

- Generate a number of file copies
- Dedicated species, reactions, and kinetic parameters can be marked as fixed during evolution

Fitness Evaluation

Specification of dynamical behaviour

- Input/output table: desired course of input and output species at discrete points in time
- Distinction between finite number of cases (runs) in input/output table
- Penalties can be set

```
# Initial input concentrations
# Starting with * sets the concentr
# Only one number means the concent
# Case 0
* 0
* 0
# Case 1
* 0
* 10
# Case 2
* 10
* 0
# Case 3
* 10
* 10

# Now the output data comes
# Case 0
0
# Case 1
10
# Case 2
```

Fitness evaluation

- Numerical integration of reaction network using ODE solver (SOSlib)
- Currently, mass-action kinetics
- Fitness measure given by weighted squared distance to target time course (output species)
- Minimisation of fitness value (!)

Fitness Evaluation

Specification of dynamical behaviour

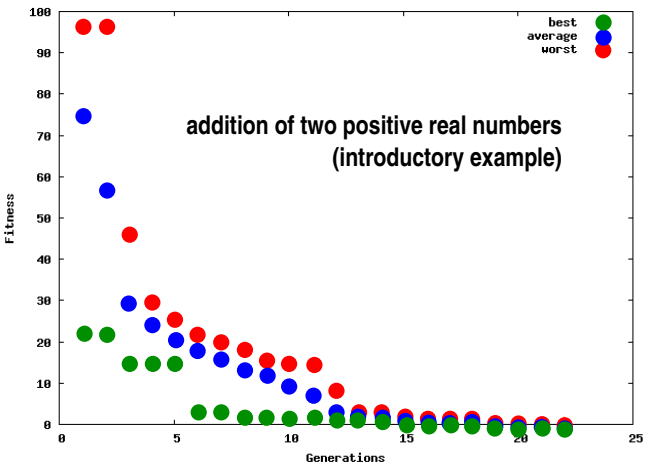
- Input/output table: desired course of input and output species at discrete points in time
- Distinction between finite number of cases (runs) in input/output table
- Penalties can be set

```
# Initial input concentrations
# Starting with * sets the concentr
# Only one number means the concent
# Case 0
* 0
* 0
# Case 1
* 0
* 10
# Case 2
* 10
* 0
# Case 3
* 10
* 10

# Now the output data comes
# Case 0
0
# Case 1
10
# Case 2
```

Fitness evaluation

- Numerical integration of reaction network using ODE solver (SOSlib)
- Currently, mass-action kinetics
- Fitness measure given by weighted squared distance to target time course (output species)
- Minimisation of fitness value (!)



fitness development (best, average, worst)

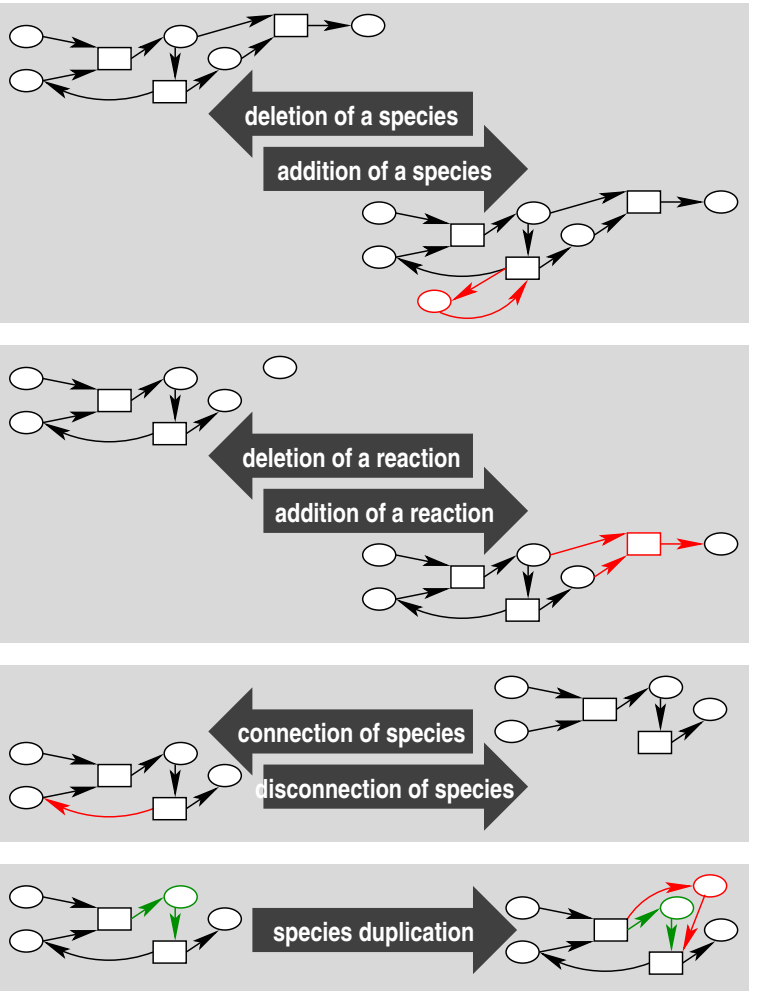
Mutation Operators

Seven mutations available, randomly selected

- Addition/deletion of a species
- Addition/deletion of a reaction
- Connection/removal of existing species to/from a reaction
- Duplication of a species with all its reactions

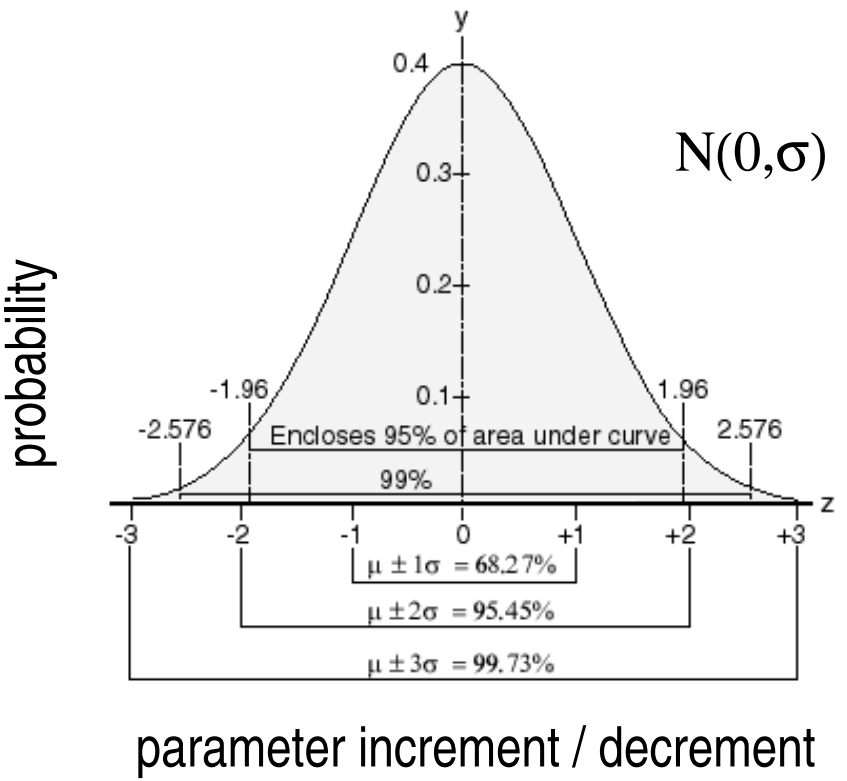
Network size can be limited.

$\Longrightarrow$ **One or several mutations per turn**



Parameter Fitting

- Adaptation of networks after structural mutation(s)
- Separate evolutionary algorithm
- Generate copies of networks resulted from structural mutation(s)
- Random selection of one or several kinetic parameters
- Mutation: addition of **Gauss variable**
- Plausibility check
- No recombination
- Environmental selection



Environmental Selection

Small population size

- Due to high computational costs of fitness evaluation

Self-adaptation of strategy parameters
(Gaussian distribution)

- Balancing between exploration of search space and fine-tuning

Non-overlapping generations

- Comma-selection supports self adaptation

Parameter settings copied from parent to offspring

- Incremental parameter fitting

Fitness proportional selection

- Combines survival of the fittest with ability to leave local optima and keeps diversity of population

Environmental Selection

Small population size

- Due to high computational costs of fitness evaluation

Self-adaptation of strategy parameters

(Gaussian distribution)

- Balancing between exploration of search space and fine-tuning

Non-overlapping generations

- Comma-selection supports self adaptation

Parameter settings copied from parent to offspring

- Incremental parameter fitting

Fitness proportional selection

- Combines survival of the fittest with ability to leave local optima and keeps diversity of population

Environmental Selection

Small population size

- Due to high computational costs of fitness evaluation

Self-adaptation of strategy parameters

(Gaussian distribution)

- Balancing between exploration of search space and fine-tuning

Non-overlapping generations

- Comma-selection supports self adaptation

Parameter settings copied from parent to offspring

- Incremental parameter fitting

Fitness proportional selection

- Combines survival of the fittest with ability to leave local optima and keeps diversity of population

Environmental Selection

Small population size

- Due to high computational costs of fitness evaluation

Self-adaptation of strategy parameters

(Gaussian distribution)

- Balancing between exploration of search space and fine-tuning

Non-overlapping generations

- Comma-selection supports self adaptation

Parameter settings copied from parent to offspring

- Incremental parameter fitting

Fitness proportional selection

- Combines survival of the fittest with ability to leave local optima and keeps diversity of population

Environmental Selection

Small population size

- Due to high computational costs of fitness evaluation

Self-adaptation of strategy parameters
(Gaussian distribution)

- Balancing between exploration of search space and fine-tuning

Non-overlapping generations

- Comma-selection supports self adaptation

Parameter settings copied from parent to offspring

- Incremental parameter fitting

Fitness proportional selection

- Combines survival of the fittest with ability to leave local optima and keeps diversity of population

Termination and Final Network Simplification

Termination

- Best fitness below configurable threshold (ideally $= 0$)
- After configurable number of generations
- After configurable number of fitness evaluations

Final network simplification

- Optional, only deletion of species keeping minimal fitness

Challenges and insufficiencies

- Premature convergence along with low diversity of population
- Overfitting (perfect replication of test cases but no further functionality of network)

Termination and Final Network Simplification

Termination

- Best fitness below configurable threshold (ideally $= 0$)
- After configurable number of generations
- After configurable number of fitness evaluations

Final network simplification

- Optional, only deletion of species keeping minimal fitness

Challenges and insufficiencies

- Premature convergence along with low diversity of population
- Overfitting (perfect replication of test cases but no further functionality of network)

Termination and Final Network Simplification

Termination

- Best fitness below configurable threshold (ideally $= 0$)
- After configurable number of generations
- After configurable number of fitness evaluations

Final network simplification

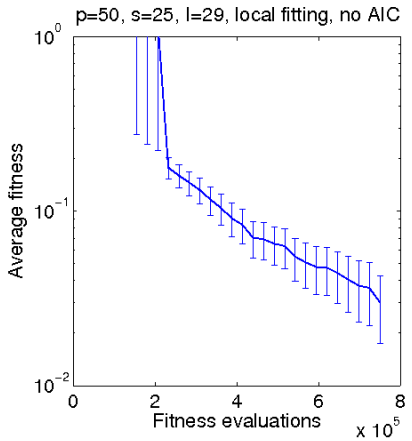
- Optional, only deletion of species keeping minimal fitness

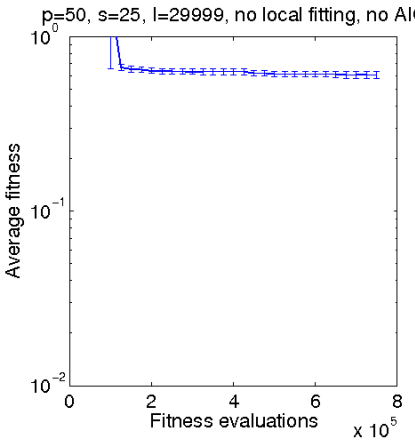
Challenges and insufficiencies

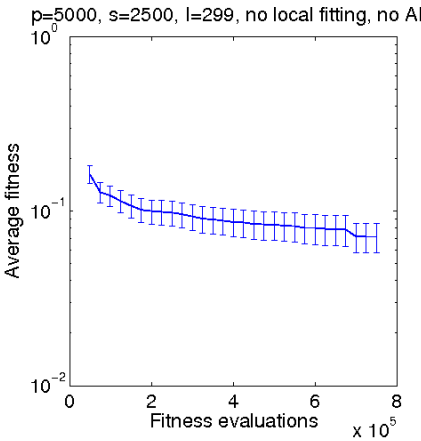
- Premature convergence along with low diversity of population
- Overfitting (perfect replication of test cases but no further functionality of network)

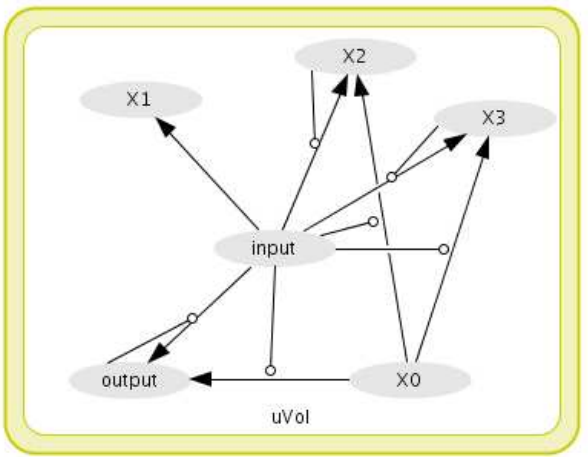
Third Root Network

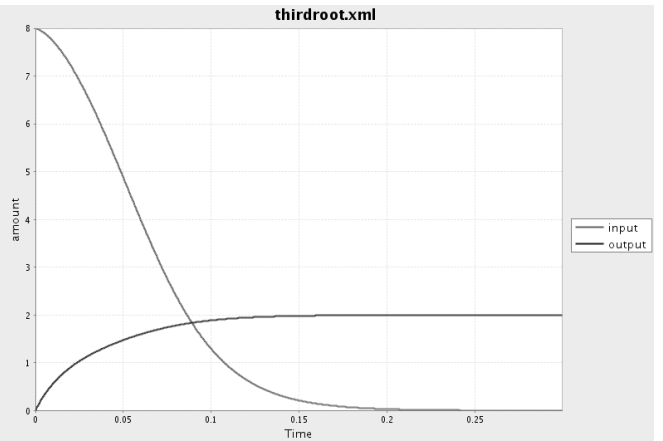
initial conc. of input species $\mapsto$ steady state conc. of output species





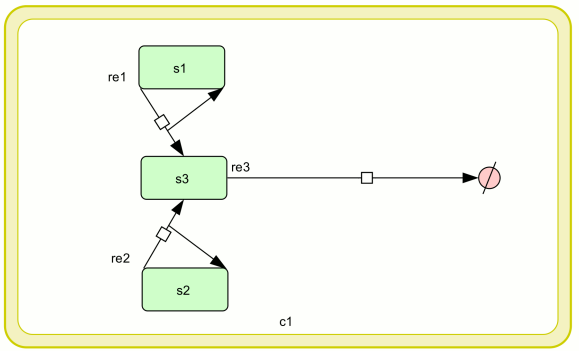


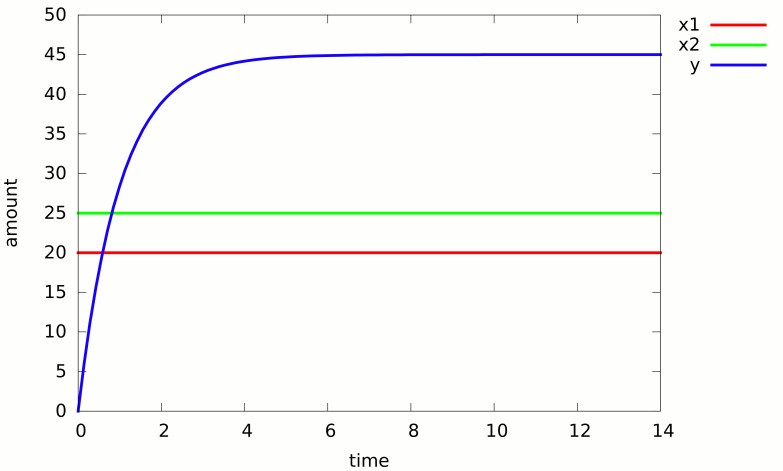




T. Lenser, T. Hinze, B. Ibrahim, P. Dittrich. Towards Evolutionary Network Reconstruction Tools for Systems Biology. In E. Marchiori, J.H. Moore, J.C. Rajapakse (Eds.), Proceedings Fifth European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Springer LNCS 4447:132-142, 2007

Addition





$$\frac{dx_1}{dt} = 0 \qquad \frac{dx_2}{dt} = 0 \qquad \frac{dy}{dt} = k_1 x_1 + k_2 x_2 - k_3 y$$

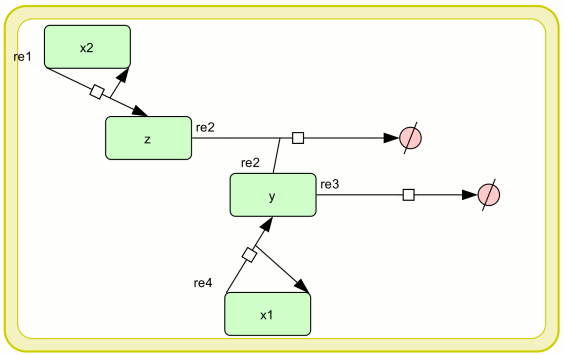
Let $k_1 = k_2 = k_3 > 0$.

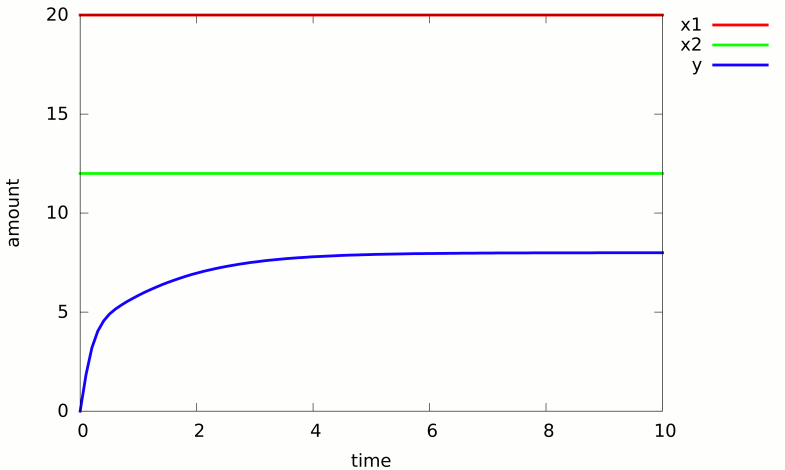
Steady state:

$$y = \lim_{t \to \infty} \left(1 - e^{-k_1 t}\right) \cdot (x_1 + x_2) = x_1 + x_2$$

B. Schau, T. Hinze, T. Lenser, I. Heiland, S. Schuster. Control System-Based Reverse Engineering of Circadian Oscillators. In I. Grosse, S. Neumann, S. Posch, F. Schreiber, P. Stadler (Eds.), Proceedings German Conference on Bioinformatics (GCB2009), p. 126-127, Martin-Luther University Halle-Wittenberg, 2009

Non-Negative Subtraction





$$\frac{dx_1}{dt} = 0 \qquad \frac{dx_2}{dt} = 0$$

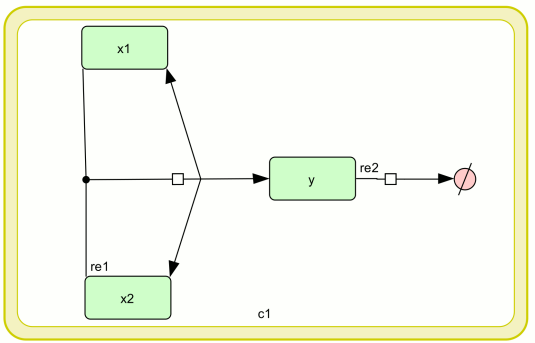
$$\frac{dy}{dt} = -k_2yz - k_1y + k_1x_1 \qquad \frac{dz}{dt} = k_1x_2 - k_2yz$$

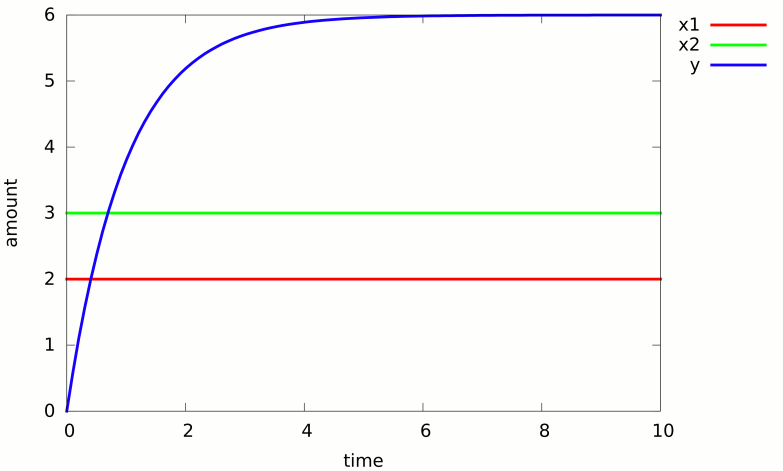
Let $k_1 > 0$ and $k_2 > 0$.

Steady state:

$$y = \begin{cases} x_1 - x_2 \text{ iff } x_1 > x_2 \\ 0 \text{ otherwise} \end{cases}$$

Multiplication





$$\frac{dx_1}{dt} = 0 \qquad \frac{dx_2}{dt} = 0 \qquad \frac{dy}{dt} = k_1 x_1 x_2 - k_2 y$$

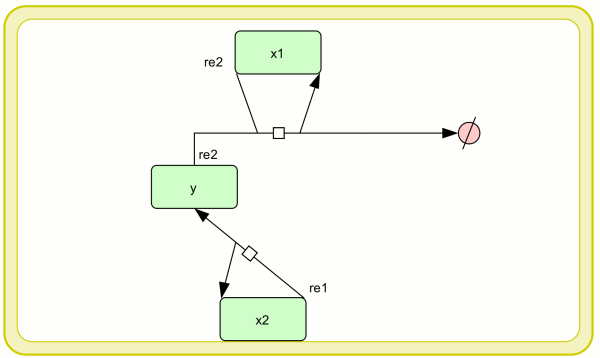
Let $k_1 = k_2 > 0$.

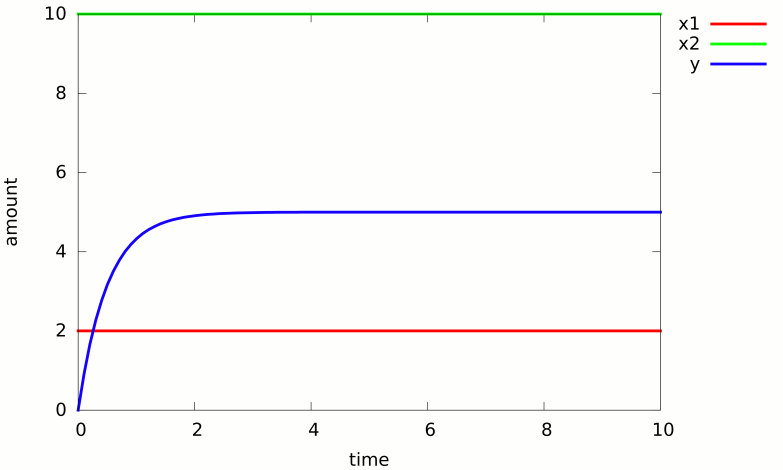
Steady state:

$$y = \lim_{t \to \infty} \left(1 - e^{-k_1 t}\right) \cdot x_1 \cdot x_2 = x_1 \cdot x_2$$

B. Schau, T. Hinze, T. Lenser, I. Heiland, S. Schuster. Control System-Based Reverse Engineering of Circadian Oscillators. In I. Grosse, S. Neumann, S. Posch, F. Schreiber, P. Stadler (Eds.), Proceedings German Conference on Bioinformatics (GCB2009), p. 126-127, Martin-Luther University Halle-Wittenberg, 2009

Division





$$\frac{dx_1}{dt} = 0 \qquad \frac{dx_2}{dt} = 0 \qquad \frac{dy}{dt} = k_2 x_2 - k_1 x_1 y$$

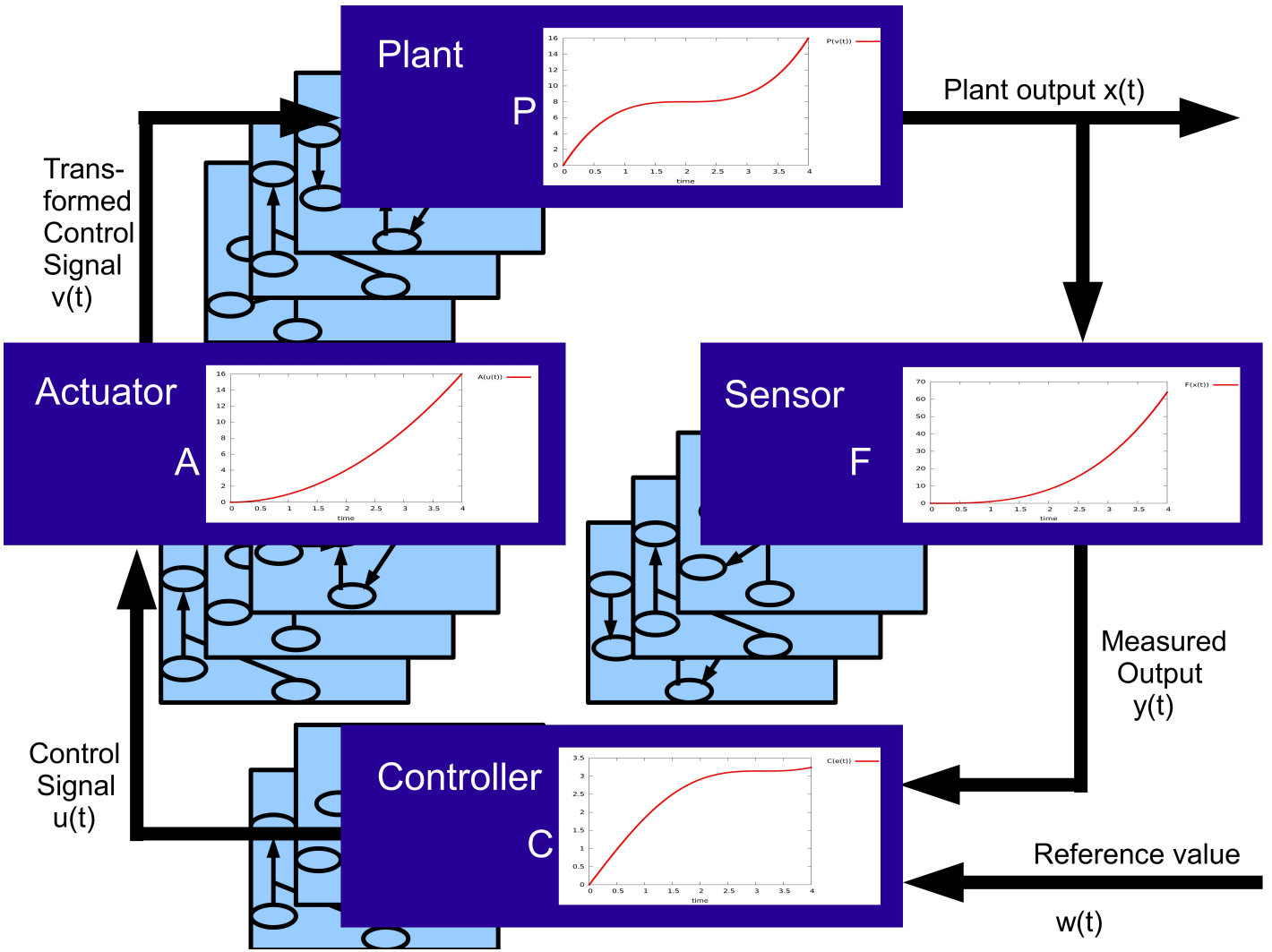
Let $k_1 = k_2 > 0$. Steady state:

$$y = \begin{cases} \lim_{t \to \infty} \left(1 - e^{-k_1 t}\right) \cdot \frac{x_2}{x_1} & \text{iff } x_1 > 0 \\ \lim_{t \to \infty} \int k_2 x_2 dt & \text{otherwise} \end{cases}$$

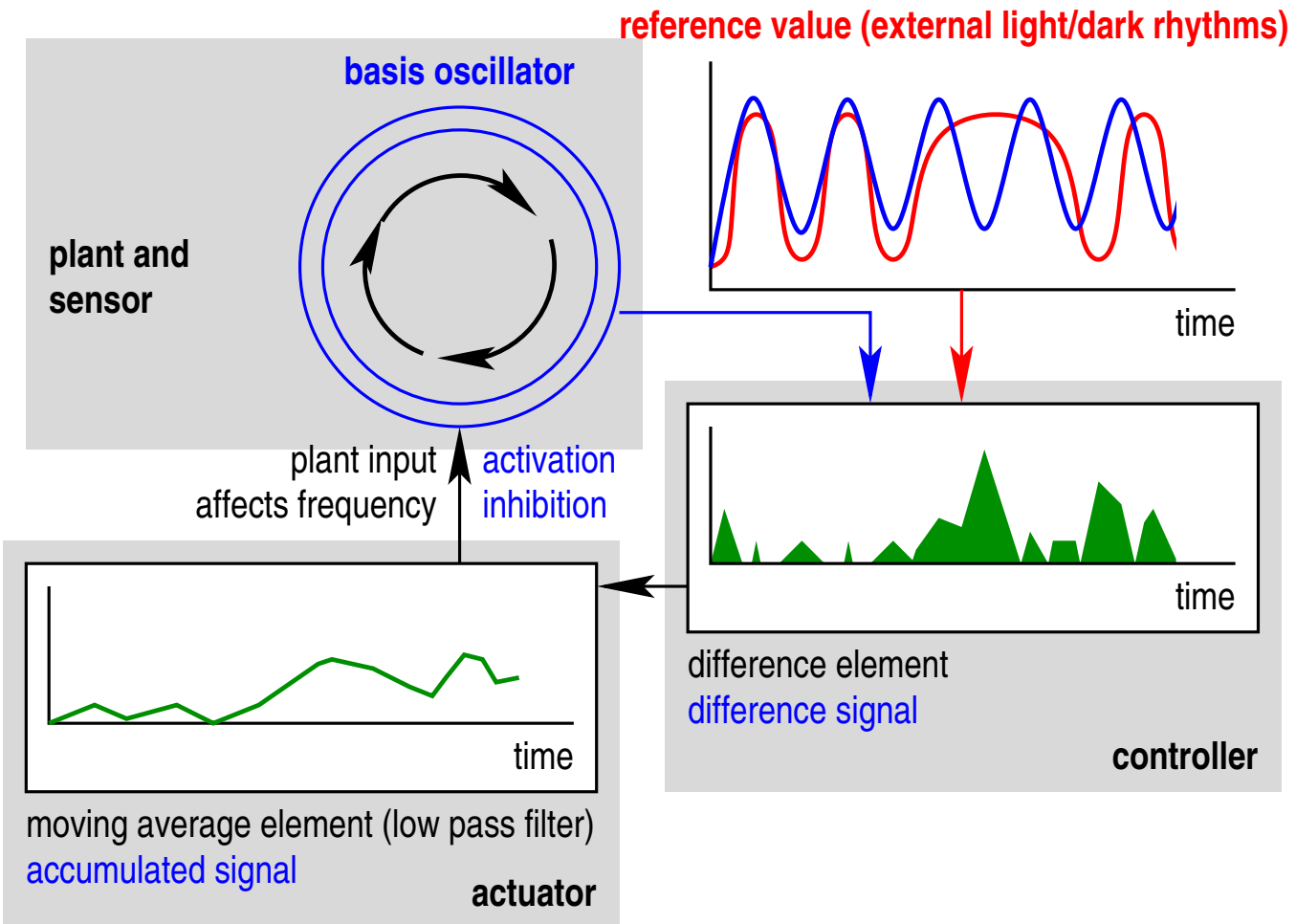
$$= \begin{cases} \frac{x_2}{x_1} & \text{iff } x_1 > 0 \\ \to \infty & \text{iff } x_1 = 0 \text{ and } x_2 > 0 \\ 0 & \text{iff } x_1 = 0 \text{ and } x_2 = 0 \end{cases}$$

Generalised Circadian System as Control System

Separation of the system into smaller functional components



Circadian Entrainment as Phase Locking Loop



Benefit from Symbiosis

membrane systems + *variable molecular structures* + *discretised reaction kinetics* . . . . . in systems biology.

Membrane systems approach Π_{CSM}

- String-objects denoted by regular expressions can manage descriptional complexity of protein binding states
- Coping with incomplete information by superpositioning of molecular configurations
- Discretised reaction kinetics enables representation of structural dynamics

Artificial reaction network evolution

- Promising heuristic approach for network reconstruction
- Exploring structural variability
- Applicable for small modules ($\leq$ 15 species), extendable by hierarchical evolution

$\Longrightarrow$ **Outlook: artificial evolution of membrane systems**

Benefit from Symbiosis

membrane systems + *variable molecular structures* + *discretised reaction kinetics* . . . . . in systems biology.

Membrane systems approach Π_{CSM}

- String-objects denoted by regular expressions can manage descriptional complexity of protein binding states
- Coping with incomplete information by superpositioning of molecular configurations
- Discretised reaction kinetics enables representation of structural dynamics

Artificial reaction network evolution

- Promising heuristic approach for network reconstruction
- Exploring structural variability
- Applicable for small modules ($\leq$ 15 species), extendable by hierarchical evolution

$\Longrightarrow$ **Outlook: artificial evolution of membrane systems**

Benefit from Symbiosis

membrane systems + *variable molecular structures* + *discretised reaction kinetics* . . . . . in systems biology.

Membrane systems approach Π_{CSM}

- String-objects denoted by regular expressions can manage descriptional complexity of protein binding states
- Coping with incomplete information by superpositioning of molecular configurations
- Discretised reaction kinetics enables representation of structural dynamics

Artificial reaction network evolution

- Promising heuristic approach for network reconstruction
- Exploring structural variability
- Applicable for small modules ($\leq$ 15 species), extendable by hierarchical evolution

$\Longrightarrow$ **Outlook: artificial evolution of membrane systems**

Benefit from Symbiosis

membrane systems + *variable molecular structures* + *discretised reaction kinetics* . . . . . in systems biology.

Membrane systems approach Π_{CSM}

- String-objects denoted by regular expressions can manage descriptional complexity of protein binding states
- Coping with incomplete information by superpositioning of molecular configurations
- Discretised reaction kinetics enables representation of structural dynamics

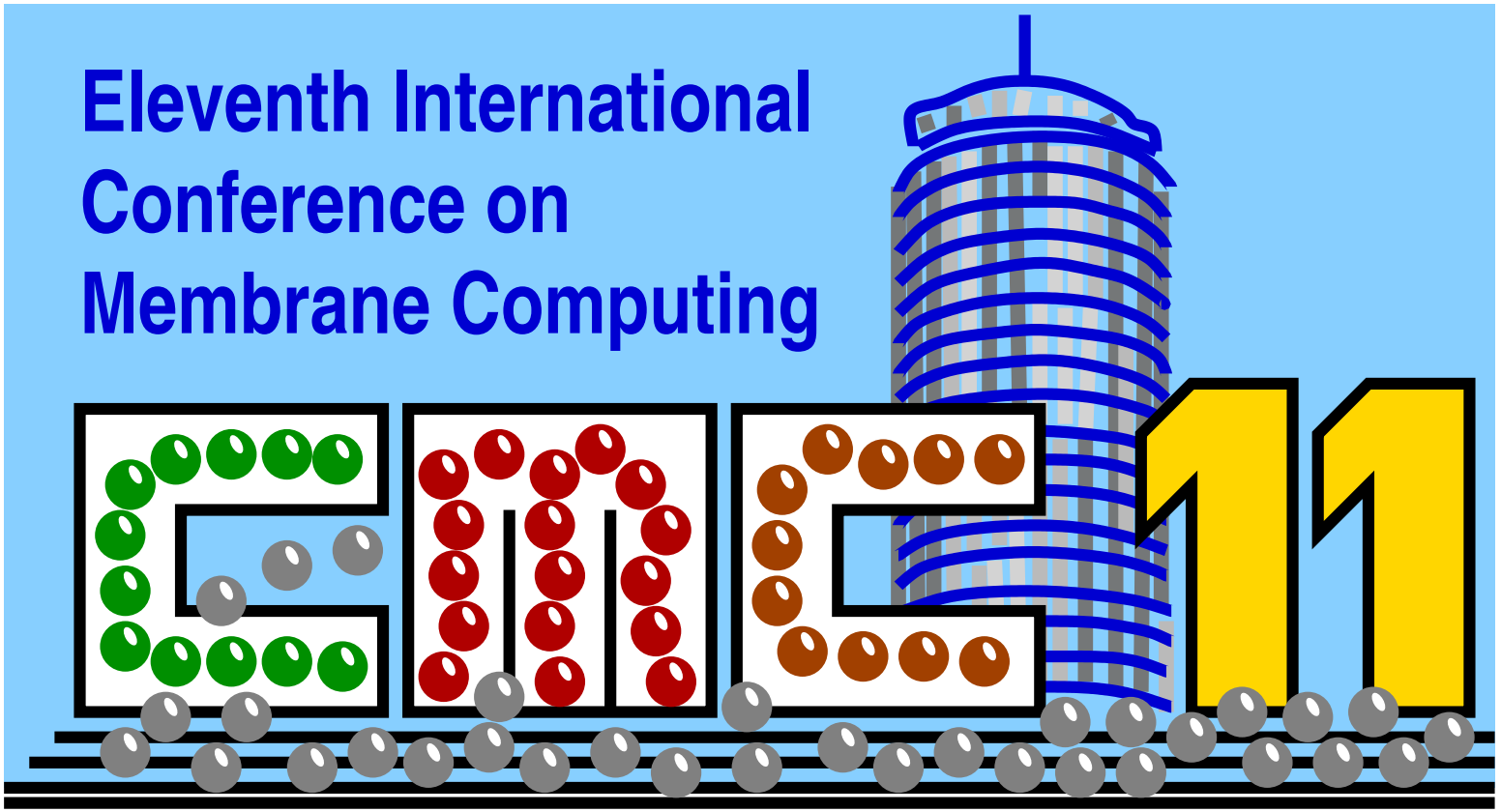
Artificial reaction network evolution

- Promising heuristic approach for network reconstruction
- Exploring structural variability
- Applicable for small modules ($\leq$ 15 species), extendable by hierarchical evolution

$\Longrightarrow$ **Outlook: artificial evolution of membrane systems**

Eleventh International Conference on Membrane Computing (CMC11)

24-27 August 2010, Jena, Germany



http://cmc11.uni-jena.de

Special Thanks go to . . .

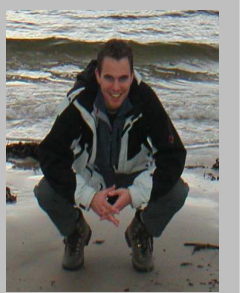
... my CMC11 coworkers

Jörn Behre
Department of Bioinformatics, FSU Jena

Gabi Escuela
Bio Systems Analysis Group, FSU Jena

Rudolf Freund
Vienna University of Technology

Thorsten Lenser
Bio Systems Analysis Group, FSU Jena



... the hosting organizations

Friedrich Schiller University of Jena (FSU)
Jena Centre for Bioinformatics (JCB)

seit1558

JCB
Jena Centre for Bioinformatics

... you for your attention. Questions?